Robust Optimization Method for Online Flowshop Problem with Uncertain Processing Times and Preemption Penalties

Mohammad Bayat^{*} and Mohammad Mahdavi Mazdeh

Department of Industrial Engineering, Iran University of Science and Technology; mohammadbayat@iust.ac.ir

Abstract

The flowshop scheduling problem has been one of the most attractive research issues. Deterministic flowshop problem is widely studied; whereas it is uncertain processing times has remained a challenge. In this paper, a robust heuristic method for this problem is presented, which is applicable whenever the lower and upper bounds of each job are available. The proposed method is capable of handling the perturbation which exists amongst the processing times. Therefore, the proposed robust method could guarantee that a small deviation of the processing times does not affect the feasibility. The performance of the proposed method is explored using some numerical examples.

Keywords: Flowshop Scheduling, Online Flowshop, Online Scheduling, Preemptive Flowshop, Robust Optimization, Uncertain Processing Times

1. Introduction

The scheduling problem aims to address the best sequence for processing of jobs such to cause the best value for the objective function. However, it is not possible to use classic models to for scheduling the jobs in stochastic and dynamic environments because unwanted and unpredicted events might possibly happen in these environments. The real environments are actually dynamic in many cases. This means that planning may be done based on conditions which do not remain constant till the end and they may experience some changes. Moreover, many of these changes are possibly beyond control of the decision maker and the decision maker could be unaware of its all details until its occurrence. Complexities of planning in such an environment are thus much more than the static environment. In other words, there are some events which must be considered in these cases including unspecified completion time of the jobs, asynchronous and online arrival of the jobs, as well as preemption

*Author for correspondence

probability and penalties of the jobs. So, it can be said that the scheduling in such an environment must be capable to make the best decision immediately after occurrence of an unpredicted event (i.e. arrival of a new job) and to give the optimal sequence according to that event. As a result, the problem under study here involves a dynamic FlowShop Scheduling (FSS).

Scheduling problem has been well studied and numerous solution strategies have been provided for various environments such as job shop and flowshop. The two-machine flowshop problem with makespan objective function and deterministic processing time can be solved optimally by Johnson's rule¹. Considering three or more machines for this problem would result in an NP-hard problem². Thus, several heuristic algorithms have been suggested for solving such problems in various studies. Some of these articles have been reviewed by Framinan et al.³. Hejazi and Saghafian reviewed 176 articles on flowshop problems with makespan objective function and various heuristic methods⁴.

The processing time in uncertain flowshop problem is unknown. This simple difference results in many complexities in uncertain problems. Makino considered a flowshop problem with two jobs and exponentially distributed processing time and developed a sequencing rule to find the schedule that minimizes the expected makespan⁵. Frostig and Adiri studied stochastic scheduling for threemachine flowshop with the objective of minimizing the distribution of schedule length6. Sethi et al suggested feedback production planning based on asymptotic analysis and computational results for a stochastic two-machine flowshop problem7. Elmaghraby and Thoney considered arbitrary processing time distributions and studied the two-machine stochastic flowshop problem⁸. In twomachine stochastic flowshop problem with exponential processing times, the expected makespan value would be minimized if the jobs are sorted non-increasingly in terms of parameter $(1/\mu_{i1}-1/\mu_{i2})$. This method was introduced by Talwar and, therefore, is known as Talwar's Rule⁹. The optimality of the method was proved later by Cunningham and Dutta¹⁰. Ku and Niu obtained an appropriate condition for stochastic dominance and illustrated that Talwar's Rule provides a stochastically minimal makespan¹¹. Soroush and Allahverdi considered total completion time criterion for stochastic two-machine flowshop scheduling problem¹². Braun et al. considered the circumstances in which the Johnson algorithm could operate optimally despite the existence of machine nonavailability intervals¹³. Laha and Chakraborty proposed an efficient stochastic hybrid heuristic for flowshop scheduling¹⁴. Portougal and Trietsch expanded Johnson's Rule and applied it to stochastic problems¹⁵. They utilized mean processing time of each job as its processing time in Johnson's Rule. Moreover, Kalczynski and Kamburowski considered Weibull distribution in Talwar's Rule¹⁶. Baker and Altheimer supposed general distributions for processing times and presented three heuristic methods for the flowshop problem with m machines. The performance of these methods was investigated through a set of problems using simulation and it was noticed that these methods had near-optimal performance¹⁷. Baker and Trietsch also considered the two-machine stochastic model with a general distribution function and explored three heuristic methods. They compared the three methods, i.e. Johnson's Heuristic method and Talwar's Heuristic method and the heuristic method of changing neighboring pairs (two neighboring jobs are considered separately and are displaced if their order can be optimized), and figured out

that none of these methods dominates the others¹⁸. They considered mean of the processing time as job processing time in both Johnson and Talwar's heuristic methods.

The other contribution of this paper is to implement the robust optimization technique to handle uncertainty in FSS problem. There are three methods for robust optimization. As the first research, Soyster proposed a model which admits the highest protection¹⁹. In his case, the objective function is determined in the worst-case condition. Ben-Tal and Nemirovski presented a more popular robust approach, which addressed a more conservative solution, where a non-linear robust optimization model is presented²⁰. Bertsimas and Sim introduced a new robust approach in which the robust counterpart is of the same class as the nominal problem and, therefore, the proposed robust approach remains linear/mixed integer, if the nominal problem is linear/mixed integer²¹. Some of the studies used robust optimization techniques in scheduling problems. Lawrence and Sewell considered the job shop problem with uncertain processing times and investigated the performance of the simple dispatching heuristics against the algorithmic solution techniques²². Kochhar et al. studied the dispatching heuristic approaches for flexible flow line scheduling23. Sabuncuoglu and Karabuk proved that the dispatching rules for interruptions are more robust compared to the optimum search algorithms for offline sequences²⁴. Wu et al. studied the increase in of stability in the single machine rescheduling problem with machine breakdowns. Their proposed method rescheduled the jobs in response to machine failures so that the minimum makespan can achieve a high scheduling stability²⁵. Lambrechts et al. developed a Tabu search algorithm to generate robust schedules, considering uncertain renewable resource availabilities²⁶. Bouyahia et al. proposed a method for the robustness design of a pre-scheduling, which assumes that the number of the jobs to be processed on parallel machines is a random variable. Their proposed method minimizes the total weighted flow time27. Ghezail et al. proposed a graphical method for responding to the disruptions in the flow shop problem²⁸. Their graphical method helps the decision maker observe the consequences of random failures and choose the best sequence.

Despite numerous studies in the last decades, a not considerable attention has been paid to preemptive online flowshop problems with uncertain processing times. In this paper a new approach has been proposed to achieve robust schedule despite uncertain processing times and unexpected arrivals of new jobs. In this problem, a number of jobs are available in the shop from the beginning of the scheduling with the remaining jobs being arrived at the shop gradually during time. In this case, an initial sequence should be selected for the existing jobs in the shop first. In addition to optimization of the objective function, this sequence must be a robust solution in the meantime, so that the minor modifications upon processing of the available jobs will not significantly impact the objective function. Meanwhile, the sequence must be selected such that a minor change during processing of a job does not alter the value of the objective function considerably. Afterwards, a re-scheduling shall be done based on every new unpredicted event. It is assumed here that the jobs arrive at the shop over the time and before the job reaches the shop, there is no information regarding time to reach, processing time, delivery time and etc. The processing times of the jobs are unspecified and demonstrate a general distribution. Only minimum and maximum processing times of a job are specified for a machine. Furthermore, after arrival of the new job and re-scheduling, the new sequence must be an optimal and robust answer of the problem. The method developed by Betrsimas and Sim was adopted in this study to establish stability among the answers²¹.

A two-step technique is suggested for this purpose: In the first step the existing jobs in the shop are scheduled at zero time using a robust solution. This initial solution is robust in comparison with the variations of the job processing times. In the next step, the proper reaction is to give one optimal sequence again when a disturbance is created (i.e. arrival of a new unpredicted job to the shop). It might be decided in this optimal sequence to stop the processing job and start another new one. In the case of preemption of a job which is already being processed, a penalty must be paid as the preemption penalty. This kind of problem has numerous examples in real world; a very simple one is a car repair shop which has customers of different characters. The repair shop can tell the customer minimum and maximum required time for processing the repair job, but the exact processing time is not known. Moreover, the repairman may even decide to leave a processing job and go to a new job when another repair order just arrives to the shop. Therefore, the time spent on the previous job might be saved (if the repairing operation involve turning, mounting/dismounting of parts and etc.). Meanwhile, it is possible that the previously implemented operation is entirely nullified and the stopped job might be needed to be started from the beginning (e.g. a job like

heating a part and etc.). The objective function includes minimization of the last job completion time on second machine (makespan).

2. Proposed Method

Many scheduling problems can be modeled based on the concepts of queuing theory. Dynamic scheduling problem of flowshop is like a queuing problem with several serial service providers. The jobs arrive at the system with a certain random distribution and receive services of the serial machines in arrangement of the flowshop. A two-machine flowshop problem is considered here with uncertain processing times. Upper and lower bounds of the processing times are known for the jobs, though their distribution is rather unspecified. It is assumed that the jobs arrive to the shop with the existing jobs there at the beginning of the scheduling being those with zero arrival time. A robust initial solution is proposed for these jobs considering the uncertain processing times of the jobs. For this purpose, robust optimization techniques are utilized. In fact, the problem deals with a dynamic flowshop problem, where arrival of a new job is deemed as an unpredicted disturbance. However, the method suggested in this paper specifies position of the new job in the previous sequence. If on arrival of the new job the operation is being done on another job, and if the new job of a higher priority than the processing one, the current processing job is stopped and postponed to another time. Therefore, preemption would be allowed during the processing, which is available in two different types:

- Preemption-Repeat: In this mode, all the operation which is done on a job until the preemption point is nullified and whenever the job is going to be resumed, all the operations must be restarted from the beginning.
- Preemption-Resume: In this mode, the operation is saved until the preemption point and it will be continued from this point when the operation is going to be resumed.

Therefore, the main issue for optimization of the problem solution is to find position of the newly arrived job in the sequence. Robust optimization is used to formulate the twomachine flowshop problem in order to mitigate the effects of variations on processing times of the jobs in the future, while the main objective was minimization of the makespan value.

2.1 Problem Assumptions and Definitions

The following assumptions shall be made:

- There are two processors with flowshop arrangement and without failure in the system. The machines work at constant and unchangeable operation speed.
- Preemption is allowed during providing the services, thus there are two modes namely preemption-repeat or preemption-resume as characteristics of any job.
- Processing time of the jobs is uncertain and with no specific distribution. Only upper and lower bounds of them are specified.
- The jobs arrive the shop during the time in an unpredicted and stochastic way.

Indexes, parameters and variables used in the proposed model are listed below:

Indexes:

- *n*: total number of jobs
- *j* : index of jobs {1, 2, ..., n}
- *i* : index of machines {1, 2}
- *k*: index of priorities {1, 2, ..., n}

Parameters:

- u_{ij} : upper bound for processing time of job *j* on machine *i*
- *l_{ij}* : lower bound for processing time of job *j* on machine *i*
- p_{ii} : processing time of job *j* on machine *i*

Variables:

- x_{ijk} : it will get 1 if job *j* on machine *i* has priority of *k*, and otherwise it will get 0
- C_{ii} : completion time of job *j* with on machine *i*
- M: a very large value

The mathematical model to minimize the makespan is given by the following relation taking into account the above-mentioned conditions for the two-machine flowshop problem:

$$\min \sum_{j=1}^{n} C_{2j} \cdot x_{2jn}$$

subject to (1)

$$\sum_{j=1}^{n} C_{ij} \cdot x_{ijk} \ge \sum_{j=1}^{n} p_{ij} x_{ijk} \quad k = 1, i = 1$$
(2)

$$\sum_{j=1}^{n} C_{ij} \cdot x_{ij(k+1)} \ge \sum_{j=1}^{n} C_{ij} \cdot x_{ijk} + \sum_{j=1}^{n} p_{ij} x_{ij(k+1)}$$
$$\forall i, k = 1, 2, 3, \dots, n-1 \quad (3)$$

$$\sum_{j=1}^{n} C_{(i+1)j} \cdot x_{ijk} \ge \sum_{j=1}^{n} C_{ij} \cdot x_{ijk} + \sum_{j=1}^{n} p_{(i+1)j} \cdot x_{ijk} \quad \forall k, i = 1 \quad (4)$$

$$\sum_{j=1}^{n} x_{ijk} = 1 \quad \forall i, \forall k$$
(5)

$$\sum_{k=1}^{n} x_{ijk} = 1 \quad \forall i, \forall j \tag{6}$$

$$x_{ijk} = \{0, 1\} \quad \forall i, \forall j, \forall k \tag{7}$$

Objective function of the above-mentioned mathematical modeling will minimize completion time of the last job on the second machine (makespan). Inequalities (2) and (3) guarantee that any job on each machine can be processed only when the job of previous priority is already finished on that machine. Inequality (4) ensures that each job on machine 2 can be done when the operation of the same job is finished in the first machine. Moreover, Inequalities (5) and (6) warrant each job to demonstrate just one priority which is devoted only to one job. Since the processing time of the jobs (p_{ij}) has some variations, a robust method is used for the mathematical model, which avoids the minor variations in the processing times of the jobs creating significant changes in value of the objective function.

To change the developed mathematical model into a robust model against variations of the processing times, the method suggested by Bertsimas and Sim was adopted. This method considers three Inequalities of (2), (3) and (4) as being robust with the technique which will be discussed.

2.2 Brief Introduction of Bertsimas and Sim Method

Consider this Inequality $\sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i$ and assume that J_i generates the set of coefficients which are defined with the restriction *i* based on the disturbance $\tilde{a}_{ij}, j \in J_i$. It is also presumed that any uncertain coefficient $\tilde{a}_{ij}, j \in J_i$ belongs to the range of $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$.

Furthermore, it is assumed that only $\lfloor \Gamma_i \rfloor$ of the parameters will be affected by the variation and disturbance so that a coefficient a_{it_i} with the value being $(\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i}$ will change as below:

$$\sum_{j} a_{ij} \cdot x_{j} + \max_{\{S_{i} \cup \{t_{i}\} \mid S_{i} \subseteq J_{i,} \mid S_{i} \mid = \lfloor \Gamma_{i} \rfloor, t_{i} \in J_{i}, t_{i} \notin S_{i}} \left\{ \sum_{j \in S_{i}} \hat{a}_{ij} \left| x_{j} \right| + \left(\Gamma_{i} - \lfloor \Gamma_{i} \rfloor \right) \hat{a}_{it_{i}} \left| x_{t_{i}} \right| \right\} \leq b_{i}$$

$$(8)$$

Where, S_i represents a subset of $\lfloor \Gamma_i \rfloor$ parameters under variation. The term which added to left hand side of the above constraint is called protection function.

2.3 Development of Robust Mathematical Model

In order to formulate the mathematical model as a robust flowshop, the protection function will be calculated for the inequalities of (2), (3) and (4) in the mathematical model. These calculations are displayed for inequalities (3) and (4) for example.

For Inequality (3) in the mathematical model:

$$\sum_{j=1}^{n} C_{ij} \cdot x_{ij(k+1)} \ge \sum_{j=1}^{n} C_{ij} \cdot x_{ijk} + \sum_{j=1}^{n} p_{ij} \cdot x_{ij(k+1)}$$

$$\Rightarrow \sum_{j=1}^{n} p_{ij} \cdot x_{ij(k+1)} \le \sum_{j=1}^{n} C_{ij} \cdot x_{ij(k+1)} - \sum_{j=1}^{n} C_{ij} \cdot x_{ijk}$$

$$\forall i, k = 1, 2, 3, \dots, n-1 \quad (9)$$

Value of the protection function is calculated as below for this inequality:

$$pf = \max_{\substack{\{s_i \cup \{t_i\} \mid s_i \in J_i \\ \mid s_i \mid = \lfloor \Gamma_i \rfloor, t_i \in J_i, t_i \notin S_i\}}} \left\{ \sum_{j \in S_i} \sum_{k=1}^{n-1} \hat{p}_{ij} \cdot x_{ij(k+1)} + \sum_{k=1}^{n-1} (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{p}_{it_i} \cdot x_{it_i(k+1)} \right\}$$
(10)

In accordance with Bertsimas and Sim method, the above-mentioned relation is equal to objective function of the mathematical model below:

$$\max \sum_{j=1}^{n} \sum_{k=1}^{n-1} \left(\hat{p}_{ij} \cdot x_{ij(k+1)} \right) \times z_{ij}$$

S.t:
$$\sum_{j=1}^{n} z_{ij} \leq \Gamma_i \quad \forall i$$

$$0 \leq z_{ij} \leq 1 \quad \forall i, \forall j$$
(11)

Based on this method, dual problem shall be calculated as below to render the problem linear:

$$\min \sum_{j=1}^{n} P_{ij} + \Gamma_{i} \cdot z_{i}$$

S t:

$$P_{ij} + z_{i} \ge \sum_{k=1}^{n-1} \hat{p}_{ij} \cdot x_{ij(k+1)} \quad \forall i, \forall j$$

$$P_{ij} \ge 0 \quad \forall i, \forall j$$

$$z_{i} \ge 0 \quad \forall i$$
(12)

Finally, objective function of dual problem is added to constraint (3) with its own constraints being added to those of the problem. Thus, inequality (3) in the robust mathematical model can be described as below:

$$\sum_{j=1}^{n} p_{ij} \cdot x_{ij(k+1)} + \sum_{j=1}^{n} P_{ij} + \Gamma_i \cdot z_i \le \sum_{j=1}^{n} C_{ij} \cdot x_{ij(k+1)} - \sum_{j=1}^{n} C_{ij} \cdot x_{ijk}$$
$$\forall i, k = 1, 2, 3, \dots, n-1 \quad (13)$$

These calculations are found for constraint (4) in a similar way:

$$pf = \max_{\substack{\{S_{(i+1)} \cup \{t_i\} \mid S_{(i+1)} \subseteq J_{(i+1)} \in J_{(i+1)} \\ \mid S_{(i+1)} \mid = \lfloor \Gamma_{(i+1)} \rfloor, t_{(i+1)} \in J_{(i+1)}, t_{(i+1)} \notin S_{(i+1)} \}} \left\{ \sum_{J \in S_{(i+1)}} \sum_{k=1}^{n} \hat{p}_{(i+1)} \cdot x_{(i+1)jk} \right. \\ \left. + \sum_{k=1}^{n} \left(\Gamma_{(i+1)} - \lfloor \Gamma_{(i+1)} \rfloor \right) \hat{p}_{(i+1)t_{i+1}} \cdot x_{(i+1)t_{i+1}k} \right\}$$
(14)

This constraint is equivalent to the following model:

$$\max \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\hat{p}_{(i+1)j} \cdot x_{(i+1)jk} \right) \times z_{(i+1)j}$$

S t:
$$\sum_{j=1}^{n} z_{(i+1)j} \leq \Gamma_{(i+1)} \quad i = 1$$

$$0 \leq z_{(i+1)j} \leq 1 \quad i = 1, \forall j$$
(15)

Dual of the above-mentioned mathematical model is obtained as follows:

$$\min_{(i=1)} \sum_{j=1}^{n} P_{(i+1)j} + \Gamma_{(i+1)} \cdot z_{(i+1)}$$
S t:

$$P_{(i+1)j} + z_{(i+1)} \ge \sum_{k=1}^{n} \hat{p}_{(i+1)j} \cdot x_{(i+1)jk} \quad \forall j, i = 1$$

$$P_{(i+1)j} \ge 0 \quad \forall j, i = 1$$

$$z_{(i+1)} \ge 0 \quad i = 1$$
(16)

The exactly same calculations are done for constraint (2) with the robust mathematical model of the problem being extracted as below:

$$\min_{j=1}^{n} C_{2j} \cdot x_{2jn} = \begin{cases} \sum_{j=1}^{n} p_{ij} x_{ijk} + \sum_{j=1}^{n} P_{ij} + \Gamma_{i} \cdot z_{i} \leq \sum_{j=1}^{n} C_{ij} \cdot x_{ijk} & k = 1, i = 1 \\ P_{ij} + z_{i} \geq \hat{p}_{ij} \cdot x_{ijk} & \forall j, k = 1, i = 1 \\ \sum_{j=1}^{n} p_{ij} \cdot x_{ij(k+1)} + \sum_{j=1}^{n} P_{ij} + \Gamma_{i} \cdot z_{i} \leq \sum_{j=1}^{n} C_{ij} \cdot x_{ij(k+1)} - \sum_{j=1}^{n} C_{ij} \cdot x_{ijk} & \forall i, k = 1, 2, 3, ..., n - 1 \\ P_{ij} + z_{i} \geq \sum_{k=1}^{n-1} \hat{p}_{ij} \cdot x_{ij(k+1)} & \forall i, \forall j & \\ \sum_{k=1}^{n} x_{ijk} = 1 & \forall i, \forall j & \\ P_{ij} \geq 0 & \forall i, \forall j, \forall k & \\ P_{ij} \geq 0 & \forall i, \forall j, \forall k & \\ P_{ij} \geq 0 & \forall i, \forall j & \\ u_{ij} \geq 0 & \forall i, \forall j & \\ u_{ij} \geq 0 & \forall i, \forall j & \\ \sum_{i=1}^{n} p_{(i+1)j} \cdot x_{ijk} + \sum_{j=1}^{n} P_{(i+1)j} + \Gamma_{(i+1)} \cdot z_{(i+1)} \leq \sum_{j=1}^{n} C_{(i+1)j} \cdot x_{(i+1)jk} - \sum_{j=1}^{n} C_{ij} \cdot x_{ijk} & \forall k, i = 1 \\ P_{(i+1)j} + z_{(i+1)} \geq \sum_{k=1}^{n} \hat{p}_{(i+1)j} \cdot x_{(i+1)jk} & \forall j, \forall k & \\ \forall j, i = 1 & \\ \sum_{j=1}^{n} x_{ijk} = 1 & \forall i, \forall j & \\ \end{cases}$$

2.4 Performance of Proposed Method upon Arrival of the New Job

It is assumed that this mathematical model will finally give an optimal and robust sequence for the existing jobs in the shop. Now, it is still possible that during implementation of the operation on the jobs based on the initial sequence, a new job arrives to the shop. There is no prediction about characteristics of this job before it arrives to the shop. In fact, arrival of a new job is accounted for an unpredicted disturbance, thus the proposed approach must perform an appropriate reaction against prioritization of the recent job. If the new job is done after end of all existing unfinished jobs of the shop, then sequence of the previous jobs will show no change. Otherwise, sequence of the other jobs will be altered as well. Thus, it can be said that the main function of the suggested method at this step is to find position of the recently introduced job. Some researchers have been launched recently to address the online scheduling in a scenario mode. After selection of a scenario, all specifications of the jobs which are going to arrive at the shop during the time in this case are mentioned, so that this problem will include much less complexity. However, online dynamic mode of the problem is considered in this study, where no data is available at all about arrival time of the new jobs before they actually arrive to the shop.

A method is presented in this research work for scheduling of this problem with the abovementioned

conditions. Since the preemption is allowed in instantaneous scheduling, and since preemption has two different types as characteristics of a job, the suggested model depends on type of the job preemption.

Assume that the initial sequence is known and a new job arrives to the shop during processing of job *a* with processing time of p_{1a} . If the remaining time of the operation is called p''_{1a} and time of the operation already performed on this job is shown with p'_{1a} , then the following equation can be written:

$$p_{1a} = p'_{1a} + p''_{1a} \tag{18}$$

Therefore, after occurrence of this disturbance (arrival of the new job), a re-scheduling must be accomplished. It is usually assumed that preemption is possible for the job under processing. Thus, since a number of the jobs are finished during the re-scheduling, total number of the unfinished jobs is shown by n' and their set is represented by j'. The job being processed can involve either two following modes in terms of preemption. Some constraints are added to the main mathematical model in each of these two modes depending on which kind of job a is and the model is run again:

2.4.1 Preemption-repeat Mode

In this mode if the job *a* is put at the first priority in the new scheduling which is presented after arrival of the new

job, it means that this job will not be preempted. Thus, only its remaining time must be considered as the maximum operation time of this job, but when the priority of this job is not first, this means that the job is preempted. So, if it is put in any other priority, the whole operation must be done on it from the beginning. This is demonstrated in terms of the two following constraints which are considered upon re-scheduling. When the time spent on job *a* is smaller than its lower bound value, the Inequality of lower bound is shown with the following relation in order to avoid a negative value for the lower bound of the job processing time:

$$p_{ij'} = p_{ij} - p'_{ij'} \cdot x_{ij'k}$$

$$p_{ij'} \le \left(u_{ij'} - p'_{ij'} \cdot x_{ij'k}\right) \quad i = 1, \ j' = a, \ k = 1$$

$$p_{ij'} \ge \max\left\{l_{ij'} - p'_{ij'} \cdot x_{ij'k}, 0\right\} \quad i = 1, \ j' = a, \ k = 1 \quad (19)$$

2.4.2 Preemption-resume Mode

In this mode, irrespective of whether the processing job is stopped or not, the time which must be spent on it is only the amount of time remained from it. Thus, this issue will be added to the mathematical model in terms of the following constraint:

$$p_{ij'} = p_{ij} = p'_{ij'}$$
$$\max\left\{ \left(l_{ij} = p'_{ij'} \right), 0 \right\} \le p_{ij'} \le \left(u_{ij} - p'_{ij} \right) \ j' = a, i = 1 \quad (20)$$

The suggested algorithm is schematically shown in Figure 1 for solving the dynamic scheduling problem of flowshop.

3. Performance Analysis of Proposed Method

To solve this problem, service provider system and jobs of the flowshop are considered as decision maker and entry of the system, respectively. The decision maker here aims to choose a sequence of jobs in any decision making situation in order to minimize the makespan obtained from sequence of jobs in the queue of service consumer. The time for decision making is when the new job arrives at the service provider system, or when the situation in which the operation is finished on the job and the service provider is ready to provide the jobs with the required services.

The jobs are generated according to GenNewJob function and arrival time to the system. The pseudo

code corresponding with the GenNewJob is defined as below:

$$[j, p_{ij}, r_{j}, mod] = GenNewJob(j)$$

$$j = j + 1;$$

$$p_{ij} = rand (l_{ij}, u_{ij});$$

$$r_{j} = rand (0, 1000);$$

$$mod = rand n\{0, 1\};$$
(21)

where, $p_{i,j}$ is time for processing time of the job *j* on machine *i* in the sequence of machines, while r_j denotes arrival time of job *j*. Moreover, rand(x, y) stands for generation of a random number in the range of (x, y) with uniform distribution. *mod* examines the preemption mode of the job as long as the job is scheduled for service. The zero value indicates the preemption-repeat mode, whereas the unit value implies a preemption-resume mode.

After completion of the operation for each job by the system, or after arrival of a new job to the system, priority of the unfinished jobs is determined by a robust mathematical model and thereafter, the operation on the job with the highest priority is started. In this case, if the just arrived job is of a higher priority in comparison with the ongoing process, then preemption will occur and preemption penalty will be incurred for the system based on characteristics of the stopped job. As can be inferred from this procedure, design and improvement of the dynamic problems are a bit different from the static ones which need a combination of simulation and optimization.

The proposed approach is implemented in MATLABR2007b software that is linked with GAMS23.6 software. Meanwhile, all tests and example problems are solved on a PC with 2.26 GHz processor and 3.00 GB of RAM. Based on the implemented robust model, first an initial sequence is created for the existing jobs in the shop. Then the just arrived jobs are simulated based on a code which is run in the MATLAB software. By using the link between these two software, after generation of a new job by MATLAB software, the implemented model is recalled in GAMS software. The final decision making is done based on information received from the newly introduced job and also the preemption mode of the processing job.

Convergence trend of simulation and optimization is shown in the Figure 2 based on the robust model for a sample problem in fifty runs.

To further investigate the results obtained from the robust model corresponding to the flowshop scheduling



Figure 1. Schematic representation of the proposed heuristic algorithm.

problem, performance of proposed method (R.O) has been studied on problems of different sizes. So, it can be said that some 20,000 problems are generated in 20 categories with different number of jobs and different specifications. Afterwards, results of the proposed method are compared with the other existing methods like FIFO, LIFO, M-SPT, M-LPT and Johnson. The amount of improvement is then statistically examined in the suggested approach as compared to the other methods. These methods are used by some companies at arrival of a new job and are briefly explained here:

FIFO & LIFO: In FIFO (First-Input-First-Output) technique, the jobs are sorted based on their arrival time to the shop. In other words, each job which has arrived to the shop earlier has a higher priority. In LIFO (Last-Input-First-Output) technique, each job which has arrived to the shop has a higher priority.



Figure 2. Convergence trend of proposed algorithm in 50 runs.

M-SPT: SPT (the Shortest Processing Time) is a scheduling pyramid in the single-machine problem, where the jobs are sorted on the basis of ascending processing times. In this case, each job with a shorter processing time has a higher priority. This rule is changed to M-SPT (Modified SPT) for the two-machine problem. For this purpose, the average processing times are calculated on machines 1 and 2 first and then, the job with smaller average processing time on machines 1 and 2 will represent a higher priority. **M-LPT:** In LPT (the Longest Processing Time), priority of the jobs with a longer processing time is higher than the others. However, in M-LPT (Modified LPT), the jobs are sorted according to descending average processing times 1 and 2. So, their priority is specified on this basis.

Johnson Method: This is a heuristic and famous method that was introduced by Johnson for the first time¹.

Other assumptions are as follows:

- 1. Arrival times of the jobs to the shop are generated via a uniform distribution in the range of [0, 1000].
- 2. If a job shows a preemption-repeat mode and the decision to stop it is taken at once, then the time spent until the preemption point is regarded as the preemption penalty.
- 3. There are 1,000 sample problems in each category, and the result of different methods are compared with each other.

4. Average lower and upper bounds of each job are taken as the real processing time for that job in order to calculate the answer for each one of the other methods.

For comparison of these methods, their performance was analyzed in the case the number of jobs is changed from 10 to 100, with some 10 categories being generated at the end. Distribution function of the processing times is unknown, but their upper and lower bounds are known. Minimum and maximum processing times of each job are defined based on the uniform distribution they have and defined in the following ranges:

$$l_{ij} \in [0, 10]$$

 $u_{ii} \in [20, 30]$ (22)

Therefore, the average lower bound of the jobs is found to be $E(l_{ij}) = 5$, while the average upper bound of them is $E(u_{ij}) = 25$. Table 1 and Figure 3 illustrate the details of this comparison. It can be well understood that performance of the developed method is noticeably better than the other conventional methods. This amount of improvement was analyzed and examined by using statistical techniques.

Performance of the proposed algorithm has also been studied when upper and lower bounds of the jobs change. For this purpose, some 10 categories were generated, when the number of jobs and the lower bound range of the processing times were assumed to remain constant (n = 50, $l_{ii} \in [0, 10]$). However, the upper bound

	1	1				
No. of jobs	R.O	Johnson	M-LPT	M-SPT	LIFO	FIFO
10	894.3	974.4	1087.7	1074.8	1373.5	1343.5
20	951.6	1115.4	1091.8	1213.8	1388.0	1311.9
30	963.7	1118.8	1148.2	1226.4	1476.6	1489.8
40	977.6	1150.7	1207.1	1242.9	1486.3	1461.3
50	994.7	1163.7	1271.3	1247.8	1598.4	1534.9
60	1021.5	1288.6	1292.0	1266.2	1611.5	1653.1
70	1122.5	1309.5	1363.9	1371.3	1780.7	1789.7
80	1272.7	1447.9	1538.9	1498.3	1840.3	1800.9
90	1436.2	1585.2	1683.8	1863.4	2006.4	2105.7
100	1599.6	1794.8	1802.9	1969.7	2339.6	2388.6

Table 1. Makespan values of the R.O as compared to other methods



Figure 3. Comparison of performance between R.O and other methods in terms of different number of jobs.

range of the processing time was changing. The obtained results are shown in Table 2 and Figure 4. There is also a significant different between the proposed method and other conventional techniques. As can be observed from the results the amount of makespan obtained from the proposed method is much favorable than the other existing methods.

For comparison of the method developed in this study with conventional scheduling rules of FIFO, LIFO, M-SPT, M-LPT, and Johnson methods, average, minimum and maximum values of the objective function obtained from the proposed method were considered against the minimum values obtained from the other methods. For this purpose, the problem was solved in 100 runs with total number of 50 jobs, lower bound of processing time $l_{ij} \in [0, 10]$ and upper bound of processing time $u_{ij} \in [50, 60]$ (see Figure 5). It can be seen from this figure that the value of objective function obtained from the proposed method is in average 0.71 smaller than the best results yielded from the other methods under study here. At the same time, minimum and maximum values of the objective function from the proposed method are

u _{ij}	R.O	Johnson	M-LPT	M-SPT	LIFO	FIFO
[10 20]	1055.5	1119.8	1209.3	1254.6	1542.6	1557.8
[20 30]	1076.5	1148.6	1263.3	1316.6	1629.8	1639.7
[30 40]	1101.4	1194.4	1343	1266.3	1716.7	1686.9
[40 50]	1143	1282	1353	1409.7	1835.9	1784.8
[50 60]	1212.5	1322.4	1419.3	1510.8	1884.9	1835.4
[60 70]	1321.4	1474.1	1704.6	1491.9	2110.8	2133.1
[70 80]	1453.2	1624.8	1800.9	1647.8	2252.2	2278.7
[80 90]	1614	1801.9	2087	1878.6	2321.6	2378.6
[90 100]	1788.3	1885.5	2087.2	2219.8	2501.2	2557.9
[100 110]	1953.6	2164.9	2395.7	2306.8	2802.9	2789.8

Table 2.Comparison of performance of R.O for different amountsof the upper bound processing times of the jobs



Figure 4. Trend of makespan values by increasing the processing time range of jobs.

approximately 0.58 and 0.89 smaller than the results from the other techniques.

At the end and by use of the statistical assumptions indicating advantage of the mean of R.O results over the other rules by using Fischer test at 95% confidence level, Table 3 is obtained as below:

As can be inferred from the results, there is a significant difference between the proposed method and the other existing methods. It can thus be declared that the proposed method provides a desirable performance for solving the dynamic problem of flowshop.

4. Conclusion

The online problem was investigated with uncertain times, allowed preemption and preemption penalty for a two-machine flowshop. The objective function involved minimization of the makespan. A method was proposed based on the technique developed by Bertsimas and Sim for stabilization of the results. Performance of the algorithm was examined with some example problems. Then, performance of the proposed method was compared with the five other methods considering different conditions



Figure 5. Comparison of minimum, maximum and average values of R.O with the best results of the other methods in 100 simulation runs.

Table 3.Fisher 95% individual confidence intervalsall pair-wise comparisons

NO.	Algorithm	Value	Significant difference
1	R.O VS FIFO	642	Yes
2	R.O VS LIFO	637	Yes
3	R.O VS M-SPT	228	Yes
4	R.O VS M-LPT	244	Yes
5	R.O VS Johnson	130	Yes

for the problem under study. The obtained results revealed that the proposed method provides a proper performance as compared to the other methods. This method can be utilized as a heuristic technique for problems in which processing times of the jobs are rather uncertain provided that lower and upper bounds of the processing times are known. Scheduling with the preemption penalty is a potential area of work in the field of scheduling. Especially, for the online uncertain problem, it is strictly believed that this research can be properly developed with more than two machines for different types of problems including those with setup times, job shop and open shop.

5. References

1. Johnson SM. Optimal two-and three-stage production schedules with setup times included. Nav Res Logist Q. 1954; 1(1):61-68.

- 2. Garey MR, Johnson DS, Sethi R. The complexity of flowshop and jobshop scheduling. Math Oper Res. 1976; 1(2):117–29.
- 3. Framinan JM, Gupta JND, Leisten R. A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. J Oper Res Soc. 2004; 55(12):1243–55.
- 4. Hejazi SR, Saghafian S. Flowshop-scheduling problems with makespan criterion: a review. Int J Prod Res. 2005; 43(14):2895–929.
- Makino T. On a scheduling problem. J Oper Res Soc Japan. 1965; 8:32–44.
- Frostig E, Adiri I. Three-machine flowshop stochastic scheduling to minimize distribution of schedule length. Nav Res Logist Q. 1985; 32(1):179–83.
- Sethi S, Yan H, Zhang Q, Zhou XY. Feedback production planning in a stochastic two-machine flowshop: Asymptotic analysis and computational results. Int J Prod Econ. 1993; 30:79–93.
- 8. Elmaghraby SE, Thoney KA. The two-machine stochastic flowshop problem with arbitrary processing time distributions, IIE Trans. 1999; 31(5):467–77.
- 9. Talwar PP. A note on sequencing problems with uncertain job times. J Oper Res Soc Japan. 1967; 9:93–7.
- Cunningham AA, Dutta SK. Scheduling jobs, with exponentially distributed processing times, on two machines of a flow shop. Nav Res Logist Q. 1973; 20(1):69–81.
- Ku PP-S, Niu SS-C. On Johnson's two-machine flow shop with random processing times. Oper Res. 1986; 34(1): 130–36.

- Soroush H, Allahverdi M. Stochastic two-machine flowshop scheduling problem with total completion time criterion. Int J Ind Eng Appl Pract. 2005; 12(2):159–71.
- Braun O, Lai T-C, Schmidt G, Sotskov YN. Stability of Johnson's schedule with respect to limited machine availability. Int J Prod Res. 2002; 40(17):4381–400.
- Laha D, Chakraborty UK. An efficient stochastic hybrid heuristic for flowshop scheduling. Eng Appl Artif Intell. 2007 Sep; 20(6):851–56.
- Portougal V, Trietsch D. Johnson's problem with stochastic processing times and optimal service level. Eur J Oper Res. 2006 Mar; 169(3):751–60.
- Kalczynski PJ, Kamburowski J. A heuristic for minimizing the expected makespan in two-machine flow shops with consistent coefficients of variation. Eur J Oper Res. 2006 Mar; 169(3):742–50.
- Baker KR, Altheimer D. Heuristic solution methods for the stochastic flow shop problem. Eur J Oper Res. 2012 Jan; 216(1):172–77.
- Baker KR, Trietsch D. Three heuristic procedures for the stochastic, two-machine flow shop problem. J Sched. 2010 Dec; 14(5):445–54.
- 19. Soyster AL. Technical Note—convex programming with set-inclusive constraints and applications to inexact linear programming. Oper Res. 1973; 21(5):1154–57.

- 20. Ben-Tal A, Nemirovski A. Robust solutions of linear programming problems contaminated with uncertain data. Math Program. 2000; 88(3):411–24.
- 21. Bertsimas D, Sim M. The price of robustness. Oper Res. 2004; 52(1):35–53.
- Lawrence SR, Sewell EC. Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. J Oper Manag. 1997; 15(1):71–82.
- 23. Kochhar S, Morris RJT. Heuristic methods for flexible flow line scheduling. J Manuf Syst. 1987; 6(4):299–314.
- Sabuncuoglu I, Karabuk S. Rescheduling frequency in an FMSwithuncertainprocessingtimesandunreliablemachines. J Manuf Syst. 1999; 18(4):268–83.
- 25. Wu SD, Storer RH, Pei-Chann C. One-machine rescheduling heuristics with efficiency and stability as criteria. Comput Oper Res. 1993; 20(1):1–14.
- 26. Lambrechts O, Demeulemeester E, Herroelen W. A tabu search procedure for developing robust predictive project schedules. Int J Prod Econ. 2008; 111(2):493–508.
- Bouyahia Z, Bellalouna M, Jaillet P, Ghedira K. A priori parallel machines scheduling, Comput. Ind Eng. 2010; 58(3):488–500.
- Ghezail F, Pierreval H, Hajri-Gabouj S. Analysis of robustness in proactive scheduling: A graphical approach. Comput Ind Eng. 2010; 58(2):193–98.