International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 5, 1345-1360, 2021
https://doi.org/10.33889/IJMEMS.2021.6.5.081

A Robust Expected Makespan for Permutation Flow Shop Scheduling Depending on Machine Failure Rate

Ghazwan Alsoufi
Department of Operations Research and Intelligent Techniques,
College of Computer Sciences and Mathematics, University of Mosul, Mosul, Iraq.
E-mail: ghazwan.alsoufi@uomosul.edu.iq

Manal Abdulkareem Zeidan
Department of Operations Research and Intelligent Techniques,
College of Computer Sciences and Mathematics, University of Mosul, Mosul, Iraq.
Corresponding author: manal.alabaji@uomosul.edu.iq

Lamyaa Jasim Mohammed
Department of Operations Research and Intelligent Techniques,
College of Computer Sciences and Mathematics, University of Mosul, Mosul, Iraq.
E-mail: lomuaajasem@uomosul.edu.iq

Abdellah Salhi
Department of Mathematical Sciences,
University of Essex, Colchester, UK.
E-mail: as@essex.ac.uk

(Received on March 15, 2021; Accepted on August 23, 2021)

Abstract

The environment of Flow Shop Scheduling Problems (FSSPs) to minimize the makespan of n jobs that have to be performed on m machines is considered. In real-world manufacturing systems nowadays, the uncertain circumstances to execute these jobs have an essential effect on the final scheduling scheme. This paper puts forward an integrated optimization heuristics that combine two distinct factors in flow shop scheduling. These factors are the variation in the processing times and the machine's reliability (machine failure rate), which must be considered to obtain optimal scheduling under stochastic assumptions. Two new approaches have been proposed in this work to achieve a robust expected makespan in the stochastic environment. The procedure is to add buffer time depending on the machine failure rate. Hence, the first procedure is to add buffer time to each operation in the mission according to the reliability of all machines (system reliability). The second one is to add buffer time to each operation depending on the reliability of each machine (machine reliability). For solving this problem with consideration to minimizing the expected makespan and maximizing the robustness simultaneously, the well-known (NEH) heuristic is implemented to schedule a set of jobs. Computational simulations are carried out with some well-studied problems taken from the OR-Library. Experimental results show that the proposed methods provide robust and efficient solutions. Moreover, the effects of some parameters on the optimization performance are discussed.

Keywords- Flow shop scheduling, Machine failure rate, Uncertainty, Robustness.

1. Introduction

Scheduling is considered the best tool that helps the decision-makers at industrial foundations, for example, to find the optimal sequence of the number of jobs. The main problem that the decision-makers face every day in manufactories is how to schedule a set of jobs to finish the mission early. The simple concept of flow shop scheduling is that there are n jobs that have to be performed on m machines to minimize the completion time for the last job (makespan). The procedure is to execute these jobs in an ordering sequence according to the selected scheduling plan. In other words, they assign limited resources to several jobs during a period so that the finishing time is minimum. Determination of the sequence to perform the activities on the number of machines is very important to minimize the makespan. Permutation Flow Shop Scheduling Problems (PFSSPs) have been well- addressed for deterministic cases, in which all parameters are given and known at the beginning of the scheduling and will not change over time.

Scheduling is a complex optimization problem in its own right. Stochastic flow shop scheduling with uncertain processing time is difficult because of inaccurate objective estimation, colossal search space and multiple local minima, especially NP-hardness (wang et al., 2005a). In flow shop scheduling, all jobs have the same processing line. Jobs are initially executed on the first machine, and to the second one in the same sequence, until the execution reaches the last machine. The results of this problem are in a solution space of $n!^m$ (Michael, 2018). However, solving this problem, without considering the uncertainties may lead to seriously suboptimal solutions or rescheduling. Few studies have considered the uncertain parameters that should be investigated to obtain a robust and reliable scheduling plan.

Over the years, the number of researchers interested in flow shop scheduling problems has increased. This increase is due to the technology growth, allowing faster run times and complex problems to be disbanded. There are many scheduling approaches to find the optimal sequence to perform a specific number of jobs, assuming that all parameters are known and will not change over time. Practically, the operations are usually influenced by uncertain and unexpected changes, affecting the final plan decision. Therefore, it is very substantial to study sequencing and scheduling under uncertain situations since they can cause disturbances and infeasibilities on scheduling plan, and thence the makespan. For this reason, over the last years, researchers tended to investigate this problem. See (Gourgand et al., 2000; González-Neira et al., 2017).

In order to reduce production costs and increase productivity, most manufacturing facilities attend to implement the production scheme and scheduling planning. There are a large number of unexpected disruptions that may invalidate the original schedules in real-life production. The unexpected events that accrue during the scheduling operation, such as variation in processing times, machine breakdown, newly arriving jobs and delay in job availability among others lead to scramble last-minute scheduling. However, the decision-makers try to avoid these disturbances, by either using reactive or proactive scheduling. In reactive scheduling, the problem is resolved by rescheduling the jobs if any unexpected event occurs and changes the data information. On the other hand, proactive scheduling is used to robust the initial scheduling plan to deal with any disturbance. The fundamental problem is how to avoid the increase in the makespan and the cost of the mission if any unexpected event occurs. Nonetheless, if the decision-maker chooses robust flow shop scheduling to overcome the disturbance, the makespan may be slightly more than regular scheduling.

Robust scheduling is defined in three considerations:

(i) If it has a relatively low cost than other schedules when disruption occurs and uses right-shifting as a scheduling algorithm. This definition was introduced by Jensen (2003).

(ii) If it can keep up the executing without consistency when the external events (disruption) occur. In other words, keeping the effects of a change on the overall schedule components at a minimum. See (Policella et al., 2004).

(iii) If it has a high probability of ensuring that makespan will not exceed the expected completion time. This definition was introduced by Liu et al. (2011).

In classical flow shop scheduling, the makespan will be deterministic but unrealistic, and this is because of the neglect of unexpected events. In contrast, stochastic flow shop scheduling gives immunity from these unexpected events. The fundamental factors affecting scheduling are processing times and machines. For most scheduling problems made so far, the processing time of each job on each machine has been assigned as a known real number. However, in the real world, information is often ambiguous or imprecise. The variation of processing times occurs either because of late arrival time to a machine or a stalled (tarry) machine. On the other hand, under the assumption that machines are always in a good state and available, the literature has widely addressed the scheduling problems. In terms of performance, the number of failures of machines on their age determines their status. Therefore, adopting the machine's failure rate and estimating it using statistical tools to include it as an uncertain factor in the scheduling procedure gives a coherent scheduling plan. It is evident that there is a relationship between the variation in processing times and the machine's reliability, that affect the scheduling scheme, i.e., if a machine did not operate in the given processing time, this is because there is a defect in this machine. The processing time of this operation will increase. This change in the processing times leads to a different makespan.

The paper is divided into the following sections: the literature corresponding to this paper is reviewed in Section 2. Problem description and NEH heuristic is given in Section 3. A robust expected makespan of the flow shop scheduling environment is introduced in Section 4. The proposed procedures are validated by computational experiments and are presented in Section 5. Finally, Section 6 covers the conclusions.

2. Problems under Consideration: A Literature Review

Permutation Flow Shop Scheduling Problems (PFSSPs) investigate the optimal sequence of arranging n jobs that must be performed on m machines to minimize the completion time of the last machine (makespan). Although many articles had dealt with deterministic flow shop scheduling problems to achieve different types of objectives such as mean flow time, machine idle time, mean flow time, due date fulfillment, among others, the following paragraphs will shed light on some articles that addressed the stochastic flow shop scheduling problems.

Ignall and Schrage (1965) developed a branch-and-bound algorithm for three machines flow shop scheduling problems with fuzzy processing times. Gourgand et al. (2000), submitted a review paper for stochastic flow shop scheduling for *m* machines and a set of n jobs, with given release dates to be executed. Two kinds of random events are distinguished: the variation in the processing times and the machine's breakdowns. The authors proposed an extension of the notation to consider stochastic events. Celano et al. (2003), developed a proper genetic algorithm (GA) for solving the fuzzy flow shop scheduling problems. The optimization involves two different objectives: the completion time minimization and the due date fulfillment; both the single and multi-objective configurations are considered. A new ranking criterion is proposed. Balasubramanian and Grossmann (2003) presented a non-probabilistic treatment of scheduling optimization under uncertainty. The concepts from fuzzy set theory are used to describe the imprecision and uncertainty in the task durations. Mixed-Integer Linear Programming (MILP) models are presented, derived from applying this approach to two different flow shop scheduling problems and new product development process scheduling. A tabu search is implemented and described to solve large scale problems. Temİz and Erol (2004) applied the fuzzy concept to deal with stochastic flow shop scheduling problems. Allaoui and Artiba (2004) dealt with the hybrid flow shop scheduling problems under maintenance constraints to optimize several objectives based on flow time and due date. In this model, setup, transportation times and cleaning are considered. Gourgand et al. (2005), proposed a theorem based on Markov chains and Chapman-Kolmogorov equations to compute the expected makespan of static stochastic permutation flow shop. The proposed scheme is combined with a metaheuristic based on simulated annealing for the scheduling problem. The job processing times are assumed to be exponentially distributed with a known rate. Wang et al. (2005a) employed and incorporated a hypothesis-test method, an effective methodology in statistics, into a GA to solve the stochastic flow shop scheduling problem to avoid premature convergence of the GA. The proposed approach is based on statistical performance and a hypothesis test. Wang et al. (2005b) addressed the problem of flow shop scheduling with stochastic processing times. Ordinal Optimisation (OO) and Optimal Computing Budget Allocation (OCBA) are hybridized as well as a genetic algorithm (GA) to suggest a Genetic Ordinal Optimization (GOO) approach. Swaminathan et al. (2007) studied the due dates in a flow shop for jobs that have different weights and uncertain processing times. Enforcement of a permutation schedule to varying degrees for dynamic flow shop is investigated to minimize Total Weighted Tardiness (TWT). Chaari et al. (2011), studied a hybrid flow shop scheduling problem whereas the processing time of each job for each machine at each stage is the source of uncertainty. Genetic algorithm is developed to solve the flow shop problems. Liu et al. (2011), proposed an improved genetic algorithm with a new generation scheme to maximize the probability of ensuring that makespan will not exceed the expected completion time. Huang et al. (2012), proposed an improved algorithm to search optimal sequence for flow shop scheduling problems with fuzzy processing times and fuzzy due dates. The longest common substring method combines with the random key method is proposed. Also, the longest common substring method combined with rearranging the mating method improves the search efficiency of the genetic algorithm. For application in large-sized problems, they also enhance this modified algorithm by CUDA based parallel computation. Baker and Altheimer (2012) investigated the stochastic flow shop problems and proposed heuristic methods. With simple computational requirements, three procedures are devised having been able to solve the deterministic counterpart. Nakhaeinejad and Nahavandi (2013) developed a method to solve a multi-objective flow shop scheduling in a fuzzy environment where processing times are fuzzy numbers. The objective functions are designed to minimize the makespan, the mean flow time and the machine idle time. The results show that the proposed algorithm can determine the preferable sequence by finding a nondominated solution for different degrees of satisfaction of constraints. Katragjini et al. (2013), three types of disruption are generated and employed so as to simultaneously interrupt the original schedules. Rescheduling algorithms are developed to accomplish the twofold objective of establishing a standard framework on the one hand, and proposing rescheduling methods that seek a good trade-off between schedule quality and stability on the other. Rahmani and Heydari (2014), a two-step procedure is used and introduced a proactive-reactive method. An initial robust solution is produced proactively against uncertain processing times in the first step using a robust optimization approach. In the next step, an appropriate reactive method is adopted to deal with this unexpected event when an unexpected disruption occurs. Juan et al. (2014), proposed an algorithm that combines Monte Carlo simulation with an Iterated Local Search metaheuristic. A simulation-optimization algorithm for the Permutation Flow Shop Problem with Stochastic Processing Times (PFSPST) is described. The simheuristic approach is based on the assumption that high-quality solutions for the deterministic version of the problem are likely to be high-quality solutions for the stochastic version. The use of reliability analysis techniques is proposed to analyze simulation outcomes or historical observations on the random variable representing the makespan associated with a given solution. Rahmani et al. (2014), proposed a multi-objective mixed-integer linear programming model. Release times and processing times are assumed to be stochastic variables with a normal distribution. The objective functions are minimizing three performance measures including makespan, total flow time and total tardiness. Chance Constrained Programming (CCP) approach and Fuzzy Goal Programming (FGP) are applied to deal with the stochastic parameters and multi-objective functions. Due to the complexity of the problem, an adapted genetic algorithm is implemented to solve large-sized problems. Framinan and Perez-Gonzalez (2015) addressed estimating the expected makespan for the permutation flow shop scheduling problems. The number of samples or simulation runs usually employed in the literature may not be adequate to derive robust conclusions concerning the performance of the different heuristics for a medium/large variation of the processing times. A procedure with a different number of replications is introduced so as to ensure that the percentage error in estimating the expected makespan is bounded with a high probability. They found that the most deterministic flow shop heuristics are efficiency for the stochastic flow shop problems. Gholami-Zanjani et al. (2017), proposed two approaches for solving flow shop scheduling under uncertainties of data: Robust Optimisation (RO) and fuzzy optimization. First, a deterministic mixed-integer linear programming model is introduced for the general problem. Then, its robust counterpart of the proposed model is dealt with. Afterward, the fuzzy flow shop model is developed. González-Neira et al. (2017), provided an overview of FSSP under uncertain circumstances and its role in production logistics. González-Neira et al. (2019), presented a simheuristic algorithm that hybridizes the Tabu Search metaheuristic and the Pareto Archived Evolution Strategy algorithm with a Monte Carlo Simulation process to minimize the expected tardiness and standard deviation of tardiness. Results show significant probability distributions and coefficients of variation in the objective functions and better results for expected tardiness and standard deviation of tardiness. Al-Behadili et al. (2020), proposed a multi-objective model and adapted a predictive-reactive based Biased Randomised Iterated Greedy (BRIG) approach for the permutation flow shop scheduling problem. The extensive experiments and statistical analysis demonstrate that the proposed multi-objective model is better than the other models in reducing the relative percentage deviation. Zanjani et al. (2021), developed a multi-objective Robust Mixed-Integer Linear Programming (RMILP) model where the due date and processing time are assumed uncertain. Fuzzy Goal Programming (FGP) is applied to solve this problem. CPLEX is implemented to study and validate the efficiency of the developed RMILP model. The developed model can assign jobs to available machines to obtain the best trade-off between two objectives including total tardiness and makespan under uncertain parameters. Xu et al. (2021), proposed a Genetic Algorithm based on the Relative Entropy of Fuzzy Sets (REFS_GA). The authors established a mathematical model of the multi-objective. In REFS_GA, the Pareto front is mapped to a fuzzy set, and the relational entropy coefficient of fuzzy sets is used to measure the similarity between the fuzzy sets of Pareto solutions and the ideal solution. Experimental results show that the REFS_GA has a good solution and performance than other methods. Around 100 articles about flow shop and flexible flow shop scheduling problems published were analyzed and classified.

3. Problem Description

In Permutation Flow Shop Scheduling Problems (PFSSPs), n jobs are supposed to be scheduled on m machines with identical orders to minimize the maximum completion time or makespan. The job J_i, i = 1,...,n include m operations O_{ij} , $j = 1, \dots, m$, that is $J_i = \{O_{i1}, O_{i2}, \dots, O_{im}\}$. Each operation will process on a specific machine $M_{j.}$, to take P_{ij} time unit. Furthermore, the operations for all jobs will proceed through the machines in the same order and each machine processes the n jobs in the same sequence which denoted by permutation set $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$. The PFSSP has been proved by Rinnooy Kan (1976) as NP-hard problem. Mathematically, PFSSP is expressed as n/m/F/C_{max} where n is the total number of jobs, m is the total number of machines, F refers to flow shop problem and C_{max} is a criterion for the makespan that needs to be minimum. For more details see (Conway et al., 2003). The makespan of PFSSP depending on the permutation job set π will calculate according to the following formula presented by (Akhshabi et al., 2012; Li and Yin, 2013).

$$C(\pi_1, 1) = P_{\pi_1,1}$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + P_{\pi_i,1} \qquad i = 2, \dots, n$$

$$C(\pi_1, j) = C(\pi_1, j-1) + P_{\pi_1,j} \qquad j = 2, \dots, m \tag{1}$$

$$C(\pi_i, j) = max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + P_{\pi_i,j} \quad {\scriptstyle i=2,\dots,n\,, j=2,\dots,m}$$

Then the makespan is

$$C_{max}(\pi) = C(\pi_n, m) \tag{2}$$

where $P_{\pi_i,j}$ refers to the operation processing time of job π_i on machine $M_{j.}$ and $C(\pi_i, j)$ represents the completion time of job π_i on machine M_j

The aim is to find the best arrangement for the n jobs in permutation set π which achieves the minimum makespan.

The deterministic PFSSP assumptions are categorized as follows:

(i) All jobs are ready at time zero and begin as soon as their turn comes.
(ii) Processing times are known and deterministic.
(iii) Setup times are included in processing times.
(iv) Machines are continuously available and each one can only process one job at a time.
(v) Each job can only be processed on one machine for once.
(vi) Job preemption is not allowed.
(vii) Buffer capacity between machines is infinite.
(viii) Only permutation schedules are allowed.
(ix) All jobs should proceed in the same order.

In stochastic PFSSPs, most of the deterministic assumptions are considered, but the processing times will change from deterministic to a random variable with distribution function G. Also, the machines are not always available.

3.1 NEH Heuristic

Nawaz et al. (1983) introduced the well-known heuristic algorithm (NEH) which is the best to date construction heuristic for solving the PFSSP for the deterministic flow shop scheduling case. The heuristic is described as follows:

(i) Sort the n jobs in non-increasing order of their total processing times.
(ii) Take the first two jobs and schedule them to minimize the partial makespan as if there were only these two jobs.
(iii) For k=3 to n do Step 4.
(iv) Insert the k-th job at the place, which minimizes the partial makespan among the k possible ones.

4. Robust Expected Makespan

The flow shop scheduling problems have been traditionally solved. Their solution might be unreasonable because the uncertainty effects are not taken into account. This section describes the solution for flow shop scheduling in uncertainty cases. The variation in the processing times due to the machine failure rate is considered. One of the most familiar ways to make the scheduling more robust and reliable is adding buffer time among jobs. However, if this additional time is more than needed, the completion time will be greater than the vulnerable scheduling, leading to high executing cost for this scheduling. On the other hand, if this additional time is too small, the scheduling plan will be affected, and it might need rescheduling. So it is essential to keep that in mind. The amount of buffer time should be equivalent between these two cases. This motivated us to address stochastic flow shop scheduling problems and add this slack time depending on the machine failure rate instead of the other weighted times. According to this approach, the buffer time will be more reasonable and adequate. This amount of additional time is significant to avoid any disturbances that may occur.

Many factors might change the expected processing times over and make the uncertainty cases exist. One of these factors is the failure of machines. Since the machines are a crucial factor in determining execution time in flow shop scheduling, it is natural that the consumption of these machines over time will affect computation time. If the machines have high reliability, then the actual processing time will be approximately equal to the expected one. In other words, there is no difference between the expected processing time and the actual one. If not, then the actual processing time will be different from the expected one. Although adding buffer time will increase the completion time of the mission and never decrease it, adding this buffer time according to the machine's reliability leads to more robust and reliable scheduling than adding a fixed buffer time or just considering the processing time as a stochastic variable. Thus, adding a short buffer time when the reliability of the machine or the system is high is logical and practical, and vice versa. To our best knowledge, consideration of the machine failure rate was not thoroughly addressed.

It is well known, the change in the processing times and breakdowns are the most frequently parameters studied under uncertainty case. Most of the proposed methods to minimize the makespan of the permutation flow shop scheduling problems considered that all machines in the system have the same efficiency during all the executing time and neglect the consumption of these machines before the mission begins. In other words, no failure or consumption might happen in the machines while executing the operations. Thus, it makes sense to take into account the consumption of machines during the scheduling. In this paper, stochastic processing times and machines failure rate parameters are considered to obtain robust and reliable scheduling. Reliability concepts are the best way to represent uncertainty or consumption in scheduling. From this motivation, considering the reliability concepts in scheduling is essential. The advantage of adding buffer time depending on the status of the machines is an immunization for the mission to overcome the disturbances.

Let $\mathcal{R}_\mathrm{j} = P_r(\mathrm{T_j} > \mathrm{t})$ be the reliability function corresponding to the machine j, with random variable $T_j \in R^+$ represent the time of the machine j spends in executing operations since its inception, i.e., lifetime. So that:

$$F_j(t) = 1 - \mathcal{R}_j(t) \tag{3}$$

where F_j is the probability of failure corresponding to the machine j

In this paper, $\mathrm{T_j}$ supposed to follow the exponential distribution, That is:

$$\mathcal{R}_j(t) = e^{-\lambda_j t} \tag{4}$$

where λ_j represents the failure rate of the machine j and t is the time spent executing all the previous operations on the machine j. Two robust approaches to estimate the expected makespan are proposed in the following subsections.

4.1 Robust Expected Makespan Using the Reliability of the System

It is well known that the status of the machines at manufacturing systems affects the processing times and thence the completion time of the production. This effect will increase the makespan. The ratio of this increase will be according to the status of this system. For instance, if the system is in perfect status, the increase will be minimal and vice-versa. The most famous mathematical concept that describes the status of a system is the reliability theory. In this paper, to compute the proper slack time, the reliability theory is used to add the buffer time to each operation in the system in order to the expected makespan. The machines in the permutation flow shop system are configured as a series. Then, the reliability of the system given by

$$\mathcal{R}_{sys}(t) = \prod_{j=1}^{m} \mathcal{R}_j(t) \tag{5}$$

Equation 5 leads to the probability of failure for system $F_{sys}(t) = 1 - \mathcal{R}_{sys}(t)$ of the random variable $T \subset R^{+}$.

Proposition 1

Let $\mathrm{P_{ij}}$ be the operation processing time of job $\mathrm{J_i}$, i = 1,...,n in permutation flow shop system with $\mathrm{M_j}$ machines j=1,...,m. Let $\mathrm{F_{sys}}$ be the probability of failure for the system, where the random variable $T \in R^{+}$ represents the last time that the system spent it in executing previous operations. Then the buffer time corresponding to the operation $\mathrm{O_{ij}}$ is as follow:

$$H_{ij}(t) = P_{ij}\, F_{sys}(t) \qquad \forall\ (i = 1, \dots , n\, , j = 1, \dots , m) \tag{6}$$

And the robust schedule will be according to the processing time $P_{ij}^{*}(t)$,

$$P_{ij}^{*}(t) = P_{ij} + H_{ij}(t) \tag{7}$$

Now, the $P_{ij}^{*}\ (t)$ of the operation $\mathrm{O_{ij}}$ is a function of system consumption. Increasing the processing time will be according to the time t which the system spent proceeding with other operations before the mission starts. One of the axiomatic properties in reliability is that the probability of failure for machine/system increases when t increases. According to Eq.6, the buffer time also will relatively increase. This property is one of the benefits of including reliability concepts in robust scheduling. The reliability will give reasonable buffer time according to machine/system consumption compared to only adding fixed time.

4.2 Robust Expected Makespan Using the Reliability of each Machine

Although the probability of failure for the system comes from the probability of failure for each machine in the system, the computing of the buffer times for all operations in all machines according to this system failure probability is sometimes inappropriate. Precisely, if some machines in the system have a perfect status and the processing times of the operations that will be executed in these machines is long, then the buffer time of these operations may be redundant and lead to increasing the cost of scheduling. Besides, the procedure assumed that the last executing times for all machines are the same. For these reasons, a second procedure is proposed to be more accurate when constructing a robust expected makespan and keeping the cost at a reasonable limit. In this procedure, the buffer time of each operation will be added correspondingly to each machine.

Proposition 2

Let P_{ij} be the operation processing time of job J_i, i=1,..., n in permutation flow shop system with M_j machines j=1,..., m. Let F_j be the probability of failure corresponding to machine j, where the random variable $T_j \in R^+$ represents the last time that the machine j spent executing previous operations. Then the buffer time for all operations corresponding to machine j is:

$$H_{ij}(t_j) = P_{ij}\, F_j(t_j) \;\; \forall\, (i = 1, \dots, n\,, j = 1, \dots, m) \tag{8}$$

And the robust schedule will be according to the processing times $P^*_{ij}(t_j)$,

$$P^*_{ij}(t_j) = P_{ij} + H_{ij}(t_j) \,\forall\, (i = 1, \dots, n) \tag{9}$$

According to Proposition 2 above, the robust scheduling will correspond to each machine's reliability and last executing time. All the results in section 5 below were obtained by assuming that the time is equal to 1000 units of time and the discussion of when this time change will be presented in the subsection 5.1.

5. Computational Experiments

To evaluate the performance of the proposed methods, computational simulations are carried out with some well-studied problems taken from the OR-Library. In this paper, 29 problems with different sizes of jobs and machines from two classes of PFSSP test problems are selected. The first eight problems are instances Car1, Car2 through to Car8 designed by Carlier (1978). The second 21 problems are instances Rec01, Rec03 through to Rec41 designed by Reeves (1995). These problems set comprise 21 test instances in seven combinations, ranging from 20 jobs with five machines to 75 jobs with 20 machines. For each size of a problem, three instances are provided. NEH heuristic is employed to find the makespan for all problems. One of the most common and important measure criteria used in industry is the makespan (Framinan et al., 2004).

The expected makespan found by Liu et al. (2011) for Car and Rec problems normally distributed, and (Gourgand et al., 2003) for Car and Rec exponentially distributed are compared with the proposed methods. Before comparison, the performance of the proposed methods must be validated. The validation is done by verifying the makespan obtained by NEH with the expected makespan of problems generated randomly. The processing time P^*_{ij} for each operation O_{ij} in the problems Car and Rec are randomly generated from the Normal distribution

$$P^*_{ij}(t) \sim N\left(P_{ij} + H_{ij}(t)\,, \frac{1}{10}(P_{ij} + H_{ij}(t))\right) \tag{10}$$

And from Exponential distribution

$$P^*_{ij}(t) \sim EXP\big(P_{ij} + H_{ij}(t)\,\big) \tag{11}$$

where H is defined in Eqs.6 and 8. According to Eqs.5, we have $\lambda_{sys} = \sum_{\forall\, j \in m} \lambda_j$. In this study we supposed $\lambda_j = 0.00005 \quad \forall\, j \in m$. For the simulation model, 10,000 replications have run to have a very good estimation of the expected makespan. The makespan in the result tables obtained by NEH heuristic for different considerations indicated by the following notations:

NEH is the makespan for deterministic data;

$E(C_{max})$ is the Expected makespan obtained by simulation with no reliability concept.

$E(C_{max})1$ is the Expected makespan computed by Liu et al. (2011).

$E(C_{max})2$ is the Expected makespan computed by Gourgand et al. (2003).

$E(C_{max}Sys1)$ is the Expected makespan computed by adding the ratio of system failure rate to each operation for the simulated data.

$E(C_{max}Sys2)$ is the Expected makespan obtained by the simulation resulting from combining the distribution parameters with the ratio of system failure rate.

$E(C_{max}Mac1)$ is the Expected makespan computed by adding the ratio of machine failure rate to the corresponding operation for the simulated data.

$E(C_{max}Mac2)$ is the Expected makespan obtained by the simulation resulting from combining the distribution parameters with the ratio of the corresponding machine failure rate.

Table 1. Computational results for car problems normally distributed.

Problems Information		**NEH**	**E(c_{max})**	**E(c_{max})1**	**System reliability**		**Machine reliability**	
Name	**Size**				**E(C_{max}Sys1)**	**E(C_{max}Sys2)**	**E(C_{max}Mac1)**	**E(C_{max}Mac2)**
Car1	11x5	7038	7036.17	7152.73	8592.56	8593.72	7379.32	7379.97
Car2	13x4	7376	7372.18	7534.52	8708.53	8711.51	7731.72	7733.04
Car3	12x5	7399	7399.92	7543.52	9036.70	9036.85	7760.75	7760.81
Car4	14x4	8129	8072.67	8270.02	9546.31	9549.82	8468.00	8469.14
Car5	10x6	7835	7842.13	7946.51	9875.71	9873.41	8224.73	8223.97
Car6	8x9	8773	8770.76	8893.55	11949.0	11950.3	9198.52	9198.80
Car7	7x7	6590	6598.07	6702.77	8546.56	8542.03	6919.86	6918.97
Car8	8x8	8564	8498.57	8694.23	11300.3	11315.2	8913.05	8916.89

Table 1 and Table 3 show the makespan obtained by NEH directly from the original data and the expected makespan obtained by $E(C_{max})$ and $E(C_{max})1$. The expected makespan obtained by the proposed methods is more robust and reliable than the one computed by Liu et al. (2011). This is because the increase in the expected makespan gives more reasonable scheduling than the one obtained by Liu et al. (2011) regarding the big numbers of the original data, thence the makespan. The expected makespan found by system reliability is greater than the expected makespan found by machine reliability. This is because the robust scheduling using the reliability for each machine is more sensitive than the system reliability. In the other words, the buffer time will add to each operation according to the status of its machine, while in the system reliability, the buffer time of all the operations will be according to the reliability of the system as a whole, which certainly has a higher failure rate than each machine separately.

Table 2. Computational results for car problems exponentially distributed.

Problems Information		**NEH**	**E(c_{max})**	**E(c_{max})2**	**System reliability**		**Machine reliability**	
Name	**Size**				**E(C_{max}Sys1)**	**E(C_{max}Sys2)**	**E(C_{max}Mac1)**	**E(C_{max}Mac2)**
Car1	11x5	7038	8288.639	11341.1	10121.983	10118.079	8692.746	8690.618
Car2	13x4	7376	8544.288	11076.0	10093.233	10106.948	8961.058	8974.561
Car3	12x5	7399	8941.461	12893.7	10919.291	10903.841	9377.612	9367.716
Car4	14x4	8129	9251.672	12671.6	10928.824	10921.555	9702.795	9693.147
Car5	10x6	7835	9201.599	12098.6	11586.219	11586.687	9650.215	9642.871
Car6	8x9	8773	10376.27	14417.4	14136.230	14133.150	10882.22	10888.76
Car7	7x7	6590	8092.151	10613.1	10481.868	10469.303	8486.809	8480.003
Car8	8x8	8564	9798.224	13081.6	13028.314	13040.827	10276.090	10269.80

Table 3. Computational results for rec problems normally distributed.

Problems Information		NEH	E(c_{max})	E(c_{max})1	System reliability		Machine reliability	
Name	Size				E(C_{max}Sys1)	E(C_{max}Sys2)	E(C_{max}Mac1)	E(C_{max}Mac2)
Rec01	20x05	1320	1303.48	1340.82	1591.46	1592.00	1367.05	1367.07
Rec03	20x05	1116	1129.15	1133.02	1378.67	1378.27	1184.11	1183.97
Rec05	20x05	1296	1280.18	1315.80	1563.38	1563.36	1342.59	1342.46
Rec07	20x10	1626	1635.22	1645.89	2278.65	2277.35	1714.97	1714.75
Rec09	20x10	1583	1585.28	1602.40	2209.06	2206.96	1662.64	1662.62
Rec11	20x10	1550	1529.19	1569.91	2130.38	2132.78	1603.64	1604.54
Rec13	20x15	2002	2000.37	2025.90	3055.89	3059.64	2097.94	2098.67
Rec15	20x15	2025	2041.45	2045.59	3118.71	3116.09	2141.08	2140.94
Rec17	20x15	2019	2014.72	2044.40	3077.95	3074.41	2113.05	2112.76
Rec19	30x10	2185	2190.48	*	3052.85	3050.03	2297.38	2297.19
Rec21	30x10	2131	2102.17	*	2927.93	2929.75	2204.50	2205.19
Rec23	30x10	2110	2126.46	*	2962.26	2960.75	2229.80	2229.81
Rec25	30x15	2644	2657.85	*	4059.89	4061.62	2787.50	2788.27
Rec27	30x15	2505	2514.67	*	3841.35	3840.14	2637.39	2637.75
Rec29	30x15	2391	2400.98	*	3667.96	3658.95	2518.10	2516.68
Rec31	50x10	3171	3205.61	*	4466.69	4465.17	3361.77	3362.00
Rec33	50x10	3241	3206.42	*	4468.05	4468.42	3362.58	3362.24
Rec35	50x10	3313	3318.62	*	4625.08	4625.61	3480.35	3480.34
Rec37	75x20	5284	5254.62	*	8576.14	8574.80	5510.91	5510.90
Rec39	75x20	5299	5351.21	*	8734.75	8734.28	5613.13	5612.67
Rec41	75x20	5242	5270.99	*	8602.84	8599.02	5527.82	5527.95

(*) No results found in Liu et al. (2011)

Table 4. Computational results for rec problems exponentially distributed.

Problems Information		NEH	E(c_{max})	E(c_{max})2	System reliability		Machine reliability	
Name	Size				E(C_{max}Sys1)	E(C_{max}Sys2)	E(C_{max}Mac1)	E(C_{max}Mac2)
Rec01	20x05	1320	1731.66	2018.2	2412.96	2419.43	1816.04	1817.00
Rec03	20x05	1116	1297.59	1641.7	1584.65	1588.99	1360.92	1359.32
Rec05	20x05	1296	1429.61	1915.0	1745.81	1744.46	1499.27	1497.75
Rec07	20x10	1626	1873.43	2603.6	2610.81	2615.65	1964.66	1970.31
Rec09	20x10	1583	1804.93	*	2515.14	2509.74	1893.13	1893.81
Rec11	20x10	1550	1731.66	*	2412.96	2419.43	1816.04	1817.00
Rec13	20x15	2002	2378.54	3448.4	3633.66	3637.76	2494.79	2495.75
Rec15	20x15	2025	2359.74	*	3604.97	3603.62	2474.71	2477.20
Rec17	20x15	2019	2329.46	*	3558.61	3564.70	2443.02	2442.52
Rec19	30x10	2185	2467.79	3416.3	3438.68	3434.31	2588.50	2582.54
Rec21	30x10	2131	2369.07	*	3301.02	3299.54	2484.83	2484.09
Rec23	30x10	2110	2368.89	*	3301.04	3304.65	2484.16	2486.31
Rec25	30x15	2644	3022.59	4387.9	4617.22	4611.40	3170.21	3169.54
Rec27	30x15	2505	2880.62	*	4400.91	4399.47	3021.55	3021.10
Rec29	30x15	2391	2789.04	*	4260.46	4260.51	2925.05	2926.45
Rec31	50x10	3171	3489.52	5034.0	4862.44	4869.15	3659.47	3661.19
Rec33	50x10	3241	3539.08	*	4932.11	4934.33	3711.52	3714.98
Rec35	50x10	3313	3570.47	*	4975.30	4978.64	3744.60	3743.83
Rec37	75x20	5284	5796.86	*	9460.80	9470.45	6079.67	6081.59
Rec39	75x20	5299	5877.13	*	9592.39	9597.73	6163.38	6166.51
Rec41	75x20	5242	5849.38	8762.2	9546.42	9538.88	6135.28	6136.07

(*) No results found in Gourgand et al. (2003)

Table 2 and Table 4 show the differences between the makespan obtained by NEH and the expected makespan obtained by E(C_{max}) and E(C_{max})2. This difference is due to the nature of data which are exponentially distributed. These data have a higher mean comparing with the normal distribution one. The robust makespan E(C_{max}Sys1) and E(C_{max}Mac1) for processing time exponential distributed obtaining according to propositions 1 and 2, are also verified by E(C$_{max}$Sys2) and E(C$_{max}$Mac2), receptively. It is seen that the suggested method can obtain good results.

5.1 Case Study (Car2)

The expected makespan for Car2 instance is computed by NEH heuristic for 1000 replications generated twice. The first from Normal distribution so that $P_{ij}^{*}(t)\sim N\left(P_{ij}\,,\frac{1}{10}P_{ij}\right)$, the second from Exponential distribution so that $P_{ij}^{*}(t)\sim EXP\left(P_{ij}\ \right)$. The interval [0,10k] indicates how many time units (hours, for example) that the machines work before starting to perform this scheduling, in other words, the last executing times of the machines. In Figure 1 and Figure 2 below, the blue line represents the expected makespan for different last executing times of all system (system reliability). The orange line represents the expected makespan for the same problem (Car2) for different last executing times for each machine in the system (machine reliability). The green line represents the expected makespan for the same problem used in the two previous cases for different last executing times just for one machine. In this case study, the last executing time for machine 3 is changed. These changes are used in the two distributions: Normal and Exponential.

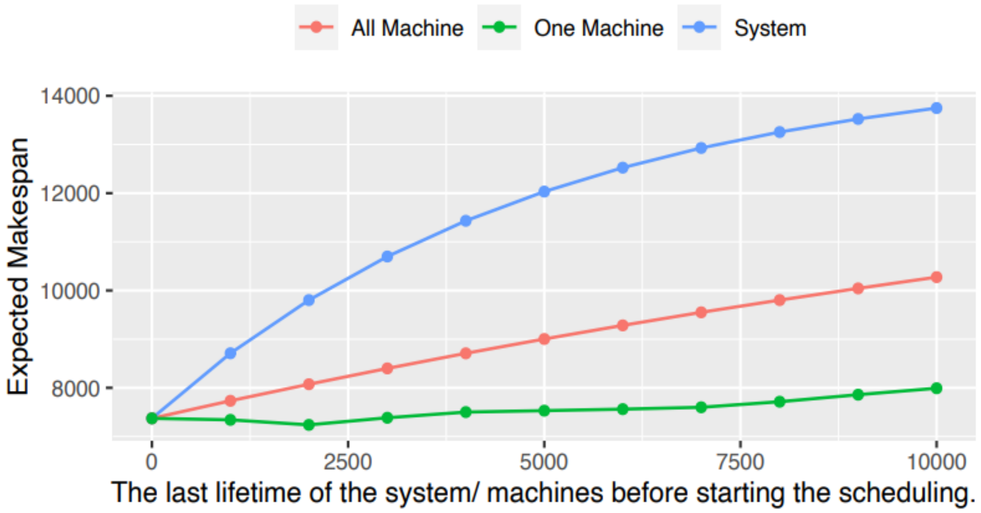

Figure 1. The Expected makespan of the system with the interval [0,10k] that indicates the last lifetime of the system/machines spent before executing the Car2 problem. In this figure $\boldsymbol{P_{ij}^{*}(t)\sim N\left(P_{ij}\,,\frac{1}{10}P_{ij}\right)}$. The blue line represents the Expected makespan computed according to the robust procedure in Proposition 1 and the orange one is devoted to illustrating the robust procedure in Proposition 2 applied to all machines in the system, while the green line is devoted to machine 3 in the system.

Observing the behavior in Figure 1 and Figure 2 above, the general behavior of the expected makespan concerning the consumption of machines of systems or machines for the two distributions is the same and is sensitive to the consumption of machines, even in one machine. The expected makespan resulted from instances generated exponentially is bigger than Normal. This is because of the behavior of exponential distribution which is always affected by time. In the two cases considered, the expected makespan based on the reliability system has a huge added buffer time, more than the machine reliability. This is justifiable because the scale parameter of the system reliability combines all the scales of the machines. i.e., $\lambda_{sys} = \sum_{\forall\, j\in m} \lambda_j$, which makes the decreasing failure rate relatively slow which, in turn, leads to an increase in the added buffer time. The drop at the beginning of the curve in the green line is normal because the new scheduling by NEH is implemented for each one in the 10000 replications generated.

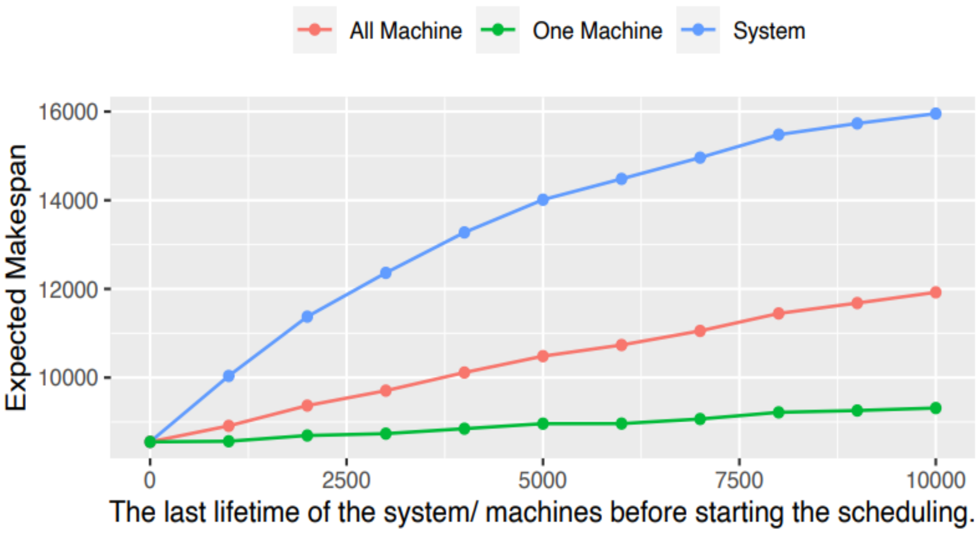

Figure 2. The Expected makespan of the system with the interval [0,10k] that indicates the last lifetime of the system/machines spent before executing the Car2 problem. In this figure $\boldsymbol{P^*_{ij}(t) \sim EXP\left(P_{ij}\right)}$. The blue line represents the Expected makespan computed according to the robust procedure in Proposition 1 and the orange one is devoted to illustrating the robust procedure in Proposition 2 applied to all machines in the system, while the green line is devoted to machine 3 in the system.

6. Conclusions

The proposed expected makespan for PFSSP considers the uncertain situations, which is the variation in processing times and the machine failure rate. This paper introduces a new mechanism for estimating a robust and reliable expected makespan for two distributions. The solutions obtained by these approaches are more reliable than traditional ones. Buffer time between jobs depending on the machine failure rate is added to overcome the variation in the processing times and the stalling machines during the mission performance. The robust and reliable expected makespan is obtained for stochastic flow shop scheduling problems by considering these two factors simultaneously. The proposed approaches are tested and validated through problems from the OR-Library. The computational results show that the proposed robust approach have promising results compared to the results obtained by Liu et al. (2011) and Gourgand et al. (2003).

Several observations are noted to be taken into account in future works. The first is to implement a multi-criterion analysis mechanism to consider bi-objective or multi-objective problems. The second is to combine the metaheuristics to minimize the makespan under uncertain circumstances. It also investigates the machine's reliability and effectiveness on the job shop scheduling problems for future research.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

All authors are very grateful to the University of Mosul-Iraq/College of Computer Sciences and Mathematics and the University of Essex-UK for their support to finish this paper.

References

Akhshabi, M., Haddadnia, J., & Akhshabi, M. (2012). Solving flow shop scheduling problem using a parallel genetic algorithm. *Procedia Technology*, *1*, 351-355.

Al-Behadili, M., Ouelhadj, D., & Jones, D. (2020). Multi-objective biased randomised iterated greedy for robust permutation flow shop scheduling problem under disturbances. *Journal of the Operational Research Society*, *71*(11), 1847-1859.

Allaoui, H., & Artiba, A. (2004). Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers & Industrial Engineering*, *47*(4), 431-450.

Baker, K.R., & Altheimer, D. (2012). Heuristic solution methods for the stochastic flow shop problem. *European Journal of Operational Research*, *216*(1), 172-177.

Balasubramanian, J., & Grossmann, I.E. (2003). Scheduling optimization under uncertainty-an alternative approach. *Computers & Chemical Engineering*, *27*(4), 469-490.

Carlier, J. (1978). Ordonnancements a contraintes disjonctives. *RAIRO-Operations Research*, *12*(4), 333-350.

Celano, G., Costa, A., & Fichera, S. (2003). An evolutionary algorithm for pure fuzzy flowshop scheduling problems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *11*(06), 655-669.

Chaari, T., Chaabane, S., Loukil, T., & Trentesaux, D. (2011). A genetic algorithm for robust hybrid flow shop scheduling. *International Journal of Computer Integrated Manufacturing*, *24*(9), 821-833.

Conway, R.W., Miller, L.W., & Maxwell, W.L. (2003). *Theory of scheduling*. Dover.

Framinan, J.M., & Perez-Gonzalez, P. (2015). On heuristic solutions for the stochastic flowshop scheduling problem. *European Journal of Operational Research*, *246*(2), 413-420.

Framinan, J.M., Gupta, J.N., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, *55*(12), 1243-1255.

Gholami-Zanjani, S.M., Hakimifar, M., Nazemi, N., & Jolai, F. (2017). Robust and fuzzy optimization models for a flow shop scheduling problem with sequence dependent setup times: A real case study on a PCB assembly company. *International Journal of Computer Integrated Manufacturing*, *30*(6), 552-563.

González-Neira, E., Montoya-Torres, J., & Barrera, D. (2017). Flow-shop scheduling problem under uncertainties: Review and trends. *International Journal of Industrial Engineering Computations*, *8*(4), 399-426.

González-Neira, E.M., Urrego-Torres, A.M., Cruz-Riveros, A.M., Henao-Garcia, C., Montoya-Torres, J.R., Molina-Sánchez, L.P., & Jimenez, J.F. (2019). Robust solutions in multi-objective stochastic permutation flow shop problem. *Computers & Industrial Engineering*, 137, 106026.

Gourgand, M., Grangeon, N., & Norre, S. (2000). A review of the static stochastic flow-shop scheduling problem. *Journal of Decision Systems*, *9*(2), 1-31.

Gourgand, M., Grangeon, N., & Norre, S. (2003). A contribution to the stochastic flow shop scheduling problem. *European Journal of Operational Research*, *151*(2), 415-433.

Gourgand, M., Grangeon, N., & Norre, S. (2005). Markovian analysis for performance evaluation and scheduling in m machine stochastic flow-shop with buffers of any capacity. *European Journal of Operational Research, 161*(1), 126-147.

Huang, C.S., Huang, Y.C., & Lai, P.J. (2012). Modified genetic algorithms for solving fuzzy flow-shop scheduling problems and their implementation with CUDA. *Expert Systems with Applications, 39*(5), 4999-5005.

Ignall, E., & Schrage, L. (1965). Application of the branch and bound technique to some flow-shop scheduling problems. *Operations Research*, *13*(3), 400-412.

Jensen, M.T. (2003). Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on Evolutionary Computation, 7*(3), 275-288.

Juan, A.A., Barrios, B.B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, *46*, 101-117.

Katragjini, K., Vallada, E., & Ruiz, R. (2013). Flow shop rescheduling under different types of disruption. *International Journal of Production Research*, *51*(3), 780-797.

Li, X., & Yin, M. (2013). A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem. *International Journal of Production Research*, *51*(16), 4732-4754.

Liu, Q., Ullah, S., & Zhang, C. (2011). An improved genetic algorithm for robust permutation flow-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *56*(1), 345-354.

Michael, L.P. (2018). *Scheduling: theory, algorithms, and systems*. Springer-Verlag, New York.

Nakhaeinejad, M., & Nahavandi, N. (2013). An interactive algorithm for multi-objective flow shop scheduling with fuzzy processing time through resolution method and TOPSIS. *The International Journal of Advanced Manufacturing Technology*, *66*(5), 1047-1064.

Nawaz, M., Enscore Jr, E.E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega, 11*(1), 91-95.

Policella, N., Oddi, A., Smith, S.F., & Cesta, A. (2004, September). Generating robust partial order schedules. In *2006 International Conference on Principles and Practice of Constraint Programming* (pp. 496-511). Springer. Berlin, Heidelberg.

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.

Rahmani, D., Ramezanian, R., & Saidi-Mehrabad, M. (2014). Multi-objective flow shop scheduling problem with stochastic parameters: fuzzy goal programming approach. *International Journal of Operational Research*, *21*(3), 322-340.

Reeves, C.R. (1995). A genetic algorithm for flowshop sequencing. *Computers & Operations Research*, *22*(1), 5-13.

Rinnooy Kan, A.H.G. (1976). *Machine scheduling problem: classification, complexity and computation*. Springer US, Nijhoff, The Hague.

Swaminathan, R., Pfund, M.E., Fowler, J.W., Mason, S.J., & Keha, A. (2007). Impact of permutation enforcement when minimizing total weighted tardiness in dynamic flowshops with uncertain processing times. *Computers & Operations Research, 34*(10), 3055-3068.

TemİZ, İ., & Erol, S. (2004). Fuzzy branch-and-bound algorithm for flow shop scheduling. *Journal of Intelligent Manufacturing*, *15*(4), 449-454.

Wang, L., Zhang, L., & Zheng, D.Z. (2005a). A class of hypothesis-test-based genetic algorithms for flow shop scheduling with stochastic processing time. *The International Journal of Advanced Manufacturing Technology*, *25*(11), 1157-1163.

Wang, L., Zhang, L., & Zheng, D.Z. (2005b). Genetic ordinal optimisation for stochastic flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *27*(1-2), 166-173.

Xu, W.J., He, L.J., & Zhu, G.Y. (2021). Many-objective flow shop scheduling optimization with genetic algorithm based on fuzzy sets. *International Journal of Production Research*, *59*(3), 702-726.

Zanjani, B., Amiri, M., Hanafizadeh, P., & Salahi, M. (2021). Robust multi-objective hybrid flow shop scheduling. *Journal of Applied Research on Industrial Engineering*, *8*(1), 40-55.