Original Research

Modelling and robust scheduling of two-stage assembly flow shop under uncertainty in assembling times

Received: 2021-08-08 / Accepted: 2021-09-29 / Published online: 2021-11-04

Maryam seyedhamzeh[1]. hossein amoozadkhalili[2] ▪ Seyed Mohammad Hassan Hosseini[3] ▪ Morteza Honarmand Azimi[4] ▪ Kamaladdin Rahmani[4]

* Corresponding Author, amoozad92@yahoo.com
1- Department of Industrial Management, Tabriz Branch, Islamic Azad University, Tabriz, Iran.
2- Department of Industrial Engineering, Nowshahr Branch, Islamic Azad University, Nowshahr, Iran.
3- Department of Industrial Engineering and Management, Shahrood University of Technology, Shahrood, Iran.
4- Department of Management, Tabriz Branch, Islamic Azad University, Tabriz, Iran

Abstract
This paper focuses on robust scheduling for an assembly flow shop where assembling times are uncertain. The considered manufacturing environment is a two-stage production system that consists of a processing stage followed by an assembly stage. There are two parallel machines in the first stage to process the components followed by an assembly stage wherein, the components are assembled to products. The majority of scheduling research considers a deterministic environment with pre-known and fixed data. However, in real-world condition, several kinds of uncertainties should be considered. In this article, the scheduling problem is tackle under uncertainty and the assembling time of each product at the second stage is the source of uncertainty. The problem is described and formulated as a mixed-integer linear programming model under deterministic condition. After that, the uncertainty issue is discussed and the robust scheduling procedure is introduced to minimize the maximum completion time of all products based on the min-max regret approach. To solve the robust scheduling an exact method is proposed to solve the problem on the small scales. Moreover, two approximate methods are modified and used to solve this NP-hard problem on the small and practical scales. The performances of the proposed methods are evaluated by several numerical examples taken from valid references. Computational results indicate that the proposed robust scheduling methods provide effective hedges against processing time uncertainty while maintaining excellent expected makespan performance.

Keywords - assembly flow shop; robustness; scheduling; two-stage; uncertainty

INTRODUCTION

The scheduling problem is one of the most important issues for shop floor managers due to its great influence on increasing the productivity of manufacturing resources. The main concern in scheduling problems is how to assign a set of limited resources to a set of jobs concerning operational constraints is obtained [1] , [2]. In the classical deterministic scheduling problem, there are a set of independent jobs to be processed by several machines in a specified format (i.e., single machine, parallel machine, flow shop, job shop, and so on). Each job has to be carried out on one of the machines during a fixed processing time, without preemption. So, the aim is to achieve the optimal schedule to satisfy one or more objectives as well as possible [3].

Many manufacturing industries produce complex products through a combination of processing and assembly structures. Therefore, integrated scheduling for the two-stage production systems includes one processing stage and one assembly stage is one of the most popular scheduling problems in industries that can provide ideal results for the related managers. In addition, adding an assembly stage to the scheduling problem makes it closer to the real-world condition [4]. It is worthy to be mentioned that, although researchers usually deal with these two stages separately, it may cause to lose ideal result [5].

Due to different kinds of scheduling problems, a broader range of optimization methods has been developed in the literature of scheduling. The majority of these proposed methods have been proposed under the traditional assumptions especially that the data are perfectly known and fixed. These assumptions allow the problem to be treated deterministically, which considerably simplifies the solution process. Nevertheless, in reality, there are several sources of uncertainty that can affect production plans and so, manufacturers are unable to provide reliable or satisfactory data for the problems that arise [6]. The presence of these uncertainty factors makes the question of modelling inevitably. This observation has highlighted the emergence of a new topic of research, called scheduling under uncertainty.

Processing and assembling time is one of the most important sources of uncertainty in scheduling problems. In accordance with the machine conditions, worker skill levels or some other manufacturing factors, the processing and assembling times of jobs and products are often uncertain in practice [7], [8]. Especially the assembling times of the final products depend directly on the tolerances, tools, and workers skills. This uncertain issue can often be tackled by some stochastic models if the probability distribution of assembling time is determined. In this way, historical data and experience are useful to obtain the probability distribution. However, due to diversifying products or developing new technology, they have neither poor historical data nor experience, which means the probability distribution is unavailable. In some other research, the scenario-based framework is applied, in which the uncertainty is modeled through the use of a number of scenarios. The scenario-based approaches use either discrete probability distributions or the discretization of continuous probability distribution functions, and the expectation of a certain performance criterion, such as the expected profit which is optimized with respect to the scheduling decision variables [9]. When decision makers are unable neither to acquire probabilistic information, nor determine the number of definite scenarios, robust framework can be useful to deal with uncertainty. The robust scheduling approaches reflect the concerns of risk-averse decision makers who may be more interested in hedging against poor system performance. This procedure focuses on hedging against the worst-case result rather than optimising expected performance under all potential condition. There are many efforts related to scheduling problems under uncertainty which used robust approach; for example, we can cite [10], [11], and [12]. This study discusses an assembly flow shop scheduling problem wherein, the assembling times of product are uncertain in the assembly stage. The objective is to minimize the maximum possible regret associated with a schedule in terms of makespan. To this end, we apply the robust approach concerning worst-case performance to the uncertain assembly flow shop scheduling problem.

The paper is organized as follows. In Section 2, we present a literature review of works related to this article. Section 3 is devoted to the problem explanation and formulation. In Section 4, we present solution approaches including an exact algorithm, a heuristic method, and a metaheuristic algorithm. Section 5 reports the data of test examples and the computational performance of the procedures. Finally, the conclusion and the direction of our future work are provided in section 6.

LITERATURE REVIEW

Since this study aims to tackle the assembly flow shop scheduling problem under uncertainty, the related studies in the literature are presented in two subsections. First, the studies dealing with the assembly flow shop scheduling problem are reviewed. After that, the search contributions focusing on the uncertainty issue in scheduling problems are investigated. Finally, to clear this paper's novelty, a comparison is provided between this research and the existing works, focusing on the common factors.

I. Literature review on the assembly flow shop scheduling problem

Integrated scheduling for two-stage production systems that consider both the processing activities and assembly operation concurrently is one of the most popular scheduling problems in industries. An assembly flow shop is a special form of the two-stage production system that consists of two types of stages: fabrication/processing or machining stage and assembly stage. Machining and assembly stages are composed of either one or a set of machines that are working in parallel [5], [13]. This problem has many applications in industry, and hence, has received increasing attention from researchers both in the field of academic research and manufacturing enterprise [14]. Lee et al. described a special case of this problem for the first time in 1993. They supposed that each product is assembled from two types of parts. The first part of each product is processed on the first machine and the second one must be processed on the second machine. Finally, the third machine assembles the two components into a product. They proved that the problem is strongly NP-complete. They also introduced a fire engine assembly plant as an application of this system [15]. After that, Potts et al. dealt with the assembly flow shop by inspiration of personal computer manufacturing [16]. The authors extended the problem in which there are m machines in the first stage and one machine in the second stage. They proposed a heuristic algorithm to solve the problem with the aim of minimizing the makespan.

Sun et al. studied the same problem as [15] and proposed a series of heuristic algorithms using the basic idea of Johnson's algorithm and Gupta's idea to solve the problem on the practical scales. The authors considered minimising the makespan as the objective function and shown that their proposed heuristics can solve all of the worst cases which cannot be solved by the existing heuristic algorithms [17]. Koulamas and Kyparisis extended the problem as [15] to three-stage assembly flow shop scheduling problem in which an intermediate operation is considered after the machining stage devoted to collecting and transporting the fabricated parts from the processing areas to the assembly area. The authors shown that the problem is strongly NP-hard and then analyzed the worst-case ratio bound for several heuristics to the problem on the large scales [18]. In similar research, Komaki et al. investigated three-stage assembly flow shop scheduling problem where the first stage has several identical parallel machines and the second and the third stages have a single machine. They proposed an improved Cuckoo Optimization Algorithm (COA) which incorporates new adjustments such as clustering and immigration of the cuckoos based on a discrete representation scheme [13]. Similarly, Maleki-Darounkolaei discussed a three-stage assembly flow shop scheduling problem with sequence-dependent setup times and blocking times between stages. They modeled the problem to minimize weighted mean completion time and makespan. In view of the NP-hard nature of the problem, the authors proposed a meta-heuristic method based on simulated annealing (SA) in order to solve the problem at hand on the small scales [19].

Allahverdi and Al-Anzi discussed a two-stage assembly flow shop scheduling problem wherein, there are m machines at the first stage and one assembly machine at the second stage [14]. The authors supposed that the setup times are treated as separate from processing times. They proved that the considered problem is strongly NP-hard, and therefore presented a dominance relation and propose three heuristics based on tabu search, a self-adaptive differential evolution (SDE), and a new self-adaptive differential evolution (NSDE) to solve the problem on the practical scales. Fattahi et al. extended the assembly flow shop in the case that the machining stage is a flexible flow shop which identical parallel machines in some machining stage followed by a single assembly machine. The authors developed an MIP model considering makespan as the objective function to formulate the problem. Due to complexity of the problem, some heuristic algorithms based on the basic idea of Johnson's rule were proposed in their paper to solve the problem on the large sizes. They also developed two tight lower bounds as reliable references to evaluate the performance of the proposed algorithms [5]. They continued their study by considering sequence-independent setup times in processing stage. The authors developed a hierarchical branch and bound algorithm just on the small-sized scales in their research [20].

Framinan et al. provided a comprehensive survey on two-stage production systems and concurrent-type scheduling methods for machining the parts and assembly the products [21]. Furthermore, a consolidated survey of assembly flow shop models was performed by Komaki et al. presented by focusing on solution methodology. They also introduced some problems receiving less attention and proposed several salient research opportunities [22].

Some studies have dealt with two-stage production systems using exact methods just for special cases. We can refer [23] and [4] who developed a branch and bound algorithm with some tight lower bounds for this problem.

Some researchers have dealt with the assembly flow shop scheduling problem with distributed machining stages. For instance we can cite Deng et al. that addressed a distributed two-stage assembly flow shop scheduling problem (DTSAFSP) and proposed a competitive memetic algorithm to minimise the makespan as the objective function. The authors also proposed an MIP model in their study to formulate the problem at hand [24]. In a similar study, Lei et al. introduced a distributed two-stage assembly flow shop scheduling problem and proposed a cooperated teaching-learning-based optimisation (CTLBO) algorithm to minimise the maximum completion time (makespan) [25]. Distributed assembly permutation flow-shop scheduling problem (DAPFSP) was also investigated by Huang et al. considering sequence-dependent set-up times. They proposed a new solution approach in which, two kinds of feasible solutions are generated in the first phase and then, a modified product insertion method is performed in the next phase. Finally, a job insertion method is used to adjust the processing order of jobs in each product. A local search method is also combined to jump out of local optima [26].

The summary of the existing literature demonstrates that scheduling for part processing and product assembly simultaneously has been increasing attention in recent years by researchers due to its application in manufacturing industries. However, many of these works have dealt with this problem under deterministic condition. There are few efforts on assembly-type scheduling problem under uncertainty that are discussed in the next subsection.

II. Literature review on scheduling problems under uncertainty

The issue of uncertainty in scheduling problems has received relatively little attention in the literature, in spite of its importance. Most of the existing efforts have assumed independent and known processing time distributions for individual jobs and proposed stochastic methods for the problem at hand [27]. In the literature, we can see other attempts that have followed a scenario-oriented framework, in which the uncertainty is modeled through the use of a number of scenarios, using either discrete probability distributions or the discretization of continuous probability distribution functions. Some works have dealt with the issue of the uncertainty in scheduling problems without any assumptions about the possible distribution of processing time or considering scenarios. These approaches are categorized as: proactive approaches, the so-called robust approaches, reactive approaches, and hybrid approaches [6]. In this section, we present the related studies that have tackled the uncertainty in scheduling problems. The term robust is often defined as a term describing a solution that does not change its performance much if uncertain parameters or unexpected events occur [28]. Billaut et al. state that a schedule is robust if its performance is relatively insensitive to the data uncertainty [29]. In this article, we consider the latter definition of robust scheduling.

Balasubramanian and Grossmann discussed multi-period flow shop scheduling problem with uncertain processing times. They proposed an MIP model for the problem to minimise the expected maximum completion time. In addition a branch and bound algorithm with an aggregated probability model was developed in their study for a special case [30]. Li and Ierapetritou performed a comprehensive review on the main approaches that have been address for the problem of production scheduling under uncertainty until 2007 [31]. Similarly, Verderameet al. provided an overview of the key contributions within the planning and scheduling communities with specific emphasis on uncertainty analysis until 2009 [32].

Allahverdi and Aydilek investigated the two-machine flow shop scheduling problem where jobs have random processing times that are bounded within certain intervals and the probability distributions of job processing times is not known. However, the lower and upper bounds of job processing times are determined. They considered minimizing the makespan as the objective function. The authors proposed several heuristic algorithms using the bounds for the problem [8]. Xu et al. dealt with the scheduling problem on identical parallel machines with uncertainty in processing time of jobs. Moreover, they supposed that the probability distribution of processing times is unknown to close the problem to real-world condition. A robust scheduling method was proposed in their study with minimal maximal deviation from the corresponding optimal schedule across all possible job-processing times. They also developed two exact algorithms to optimise this problem using a general iterative relaxation procedure [33].

A two-stage hybrid flow shop scheduling problem was addressed by Feng et al. where there is one machine at the first stage and the second stage has several identical parallel machines. The processing time of jobs in uncertain in their search with unknown probability distribution. The authors proposed a robust scheduling model the so-called, min–max regret to minimize the makespan. To this end, they first derived several properties of the worst-case scenario for a given schedule and then, developed both exact and heuristic algorithms to solve this problem [12]. A comprehensive review on the Flow-shop scheduling problem under uncertainties has been performed by González-Neira et al. that can be helpful for interested researchers [34].

Tadayonirad et al. addressed a two-stage assembly flow shop scheduling with random machine breakdowns and proposed a robust scheduling considering makespan and robustness simultaneously as two objective functions. The authors used imperialist competitive algorithm (ICA), genetic algorithm (GA), and hybridized with simulation techniques for handling complexities of the problem. Moreover, they used artificial neural network (ANN) to predict the parameters of the proposed algorithms in uncertain condition [35]. Faraji amiri and Behnamian addressed the flow shop scheduling problem under uncertainty to optimize makespan and energy consumption simultaneously. They formulated the problem and proposed a mathematical model and a scenario-based estimation of distribution algorithm (EDA) to solve the problem [36].

Recently, Wu et al. tackled a two-stage assembly flow shop scheduling problem with two scenario-dependent jobs processing times. Since the problem is NP-hard, the authors first derived a dominance property and a lower bound to propose a branch-and-bound algorithm to find a permutation schedule with minimum makespan. After that, they used Johnson's rule to propose eight polynomial heuristics for finding near-optimal solutions [37].

In addition to the assembly flow shop, there are other assembly manufacturing systems that researchers deal with them. For instance, we can cite [38] that discussed the production scheduling problem in an assembly manufacturing system with uncertain processing time and random machine breakdown. The authors formulated the problem and developed a modified master-apprentice evolutionary algorithm (MAE) for robust scheduling.

Based on the literature, the researchers considered stability and robustness to deal with the assembly-type scheduling problem during recent years. However, there is a lack of solution procedures for the assembly flow shop with uncertainty in assembling times. Therefore, we applied the robust approach concerning worst-case performance to the uncertain interval assembly flow shop scheduling problem. The considered procedure has been inspired by Feng et al. and modified for the two-stage assembly flow shop in this study.

PROBLEM DESCRIPTION

The two-stage assembly flow shop scheduling problem is a special case of the two-stage flow shop where we have m parallel machines in the first stage for processing the parts and an assembly machine in the second stage for assembling the parts to the final product. Due to the application of this problem in different manufacturing industries, many researchers have dealt with it in the last three decades. Fire engine assembly plant was introduced as an application of this production system for the first time by Lee et al. in 1993. Moreover, the production process of personal computer and body making of automotive manufacturing industries have been presented as other instances of application the assembly flow shop scheduling problem. However, a few studies have dealt with uncertainty in this problem, and this fact limits their applications in real-world manufacturing environments. The considered problem in this study is a two-stage assembly flow shop where there are two parallel machines in the first stage (processing stage) and one machine in the second stage (assembling stage). Each product is assembled from two types of components. The first component of each product must be processed on the first parallel machine and the second component is processed on the second parallel machine at the first stage. After processing and preparing two components, they are assembled into the final product in the second stage. According to the conditions of the assembly machine, different worker skill levels, and some other manufacturing factors, there is uncertainty in the assembling times of the products. We consider minimizing the maximum completion time of all products (makespan) as the objective function for this study. Figure 1 represents a schematic view of the considered problem in this study.

To clarify the problem at hand, we first formulate it under deterministic condition by proposing a mixed-integer linear programing (MIP) model. After that, the problem will be discussed under uncertainty in assembling time and is tackled using a robust approach. To complete notation and the mathematical model, we assume that part 0 is a dummy job with zero release time and zero processing time which should be placed in position one on all machines. Without loss of generality, it is also assumed that product 0 is a dummy product with zero assembly time which should be placed in position one on the assembly stage.

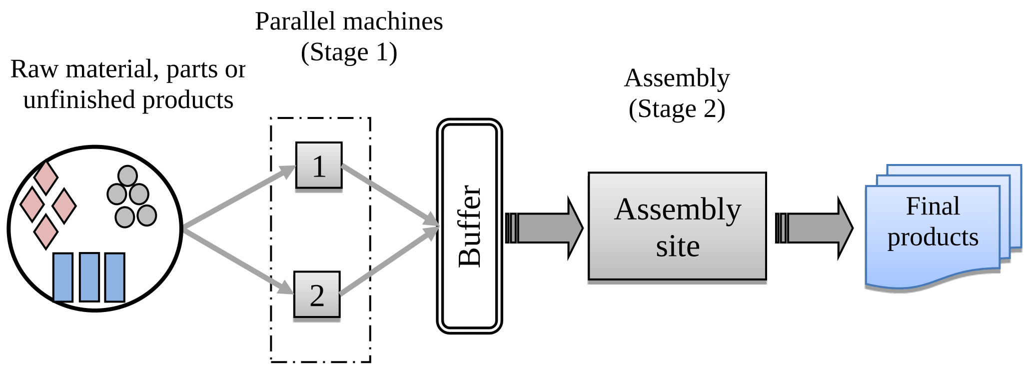

FIGURE 1
A SCHEMATIC VIEW OF THE CONSIDERED PROBLEM

- **Sets and indices**

$h, h' = \{1, 2, \dots, H\}$	Indices for products
$i, j = \{a, b\}$	Indices for parts
$m = \{A, B, C\}$	Index for processing machines

- **Parameters**

P_{ih}	Processing time of part i of product h
A_h	Assembling time of product h
L	A large number

- **Decision variables**

$X_{h,h'}$	Binary variable taking value 1 if product h is an immediate predecessor of product h' and 0 otherwise
F_h	Start time for assembly of product h
C_h	Completion time of assembling the product h
CP_{ih}	Completion time of part i of product h in processing stage
C_{max}	Maximum completion time of all products that is a continuous positive variable

- **Model formulation**

$$Min\ \ Z = Cmax \tag{1}$$

St.

$$Cmax \geq C_h \qquad \forall h \tag{2}$$

$$\sum_{h=0, h\neq h'}^{H} X_{hh'} = 1 \qquad \forall\, h' \tag{3}$$

$$\sum_{h'=1, h'\neq h}^{H} X_{hh'} \leq 1 \qquad \forall h \tag{4}$$

$$\sum_{h=1}^{H} X_{0h} = 1 \tag{5}$$

$$X_{hh'} + X_{h'h} \leq 1 \qquad \forall h, h'\ ; h \neq h' \tag{6}$$

$$CP_{ih} \geq CP_{ih'} + P_{ih} - L.(1 - X_{h'h}) \qquad \forall i, h, h'\ ; h \neq h' \tag{7}$$

$$F_h \geq CP_{ih} \qquad \forall\, i, h \tag{8}$$

$$F_h \geq C_{h'} - L.(1 - X_{h'h}) \qquad \forall h, h'\ ; h \neq h' \tag{9}$$

$$C_h \geq F_h +\ A_h \qquad \forall\, h \tag{10}$$

$$X_{hh'} \in \{0,1\} \qquad \forall h, h' \tag{11}$$

$$\mathrm{F_h, C_h, CP_{ih}, C_{max}} \geq 0 \qquad \forall i, h, h' \tag{12}$$

The above mathematical model has been proposed based on the model developed by [5] and [39]. We formulate the problem and develop the model considering uncertainty feature using relations (13) to (25).

In the mathematical model, the maximum completion time of all products (makespan) is minimized by relation (1) as the considered objective function. Constraints (2) determine the makespan. Constraints (3) and (4) ensure that every product must be exactly at one position. Equations (5) control that dummy product 0 has exactly one successor. Constraints (6) enforce the occurrence of cross-precedencies, meaning that a product cannot be at the same time both a predecessor and a successor of another product. The completion time of each part is calculated by the constraints (7). Constraints (8) and (9) determine the Start time for assembly of product h. The completion time of products is indicated by relation (10). Finally, constraints (11) and (12) are used to indicate the domains of the decision variables.

In the considered problem, the product assembling time times are assumed to be uncertain with unavailable probability distribution due to some stochastic variability. The only information is that the assembling time of an arbitrary product h lies in an interval $[A_h^{min}, A_h^{max}]$ where $0 < A_h^{min} \leq A_h^{max}]$. A scenario s is used to describe a possible product assembling family of sets where $A_h^s \in [A_h^{min}, A_h^{max}]$. All the possible scenarios are denoted as a set S.

We use Ω to demonstrate all the feasible sequences constructed from these H products. Let $F(\sigma, s)$ denotes the makespan of a sequence $\sigma\, \varepsilon\, \Omega$ under the given scenario s. Under this scenario s s, there is an optimal sequence σ^* with makespan $F_s^* = F(\sigma^*, s) = \min_{\sigma\, \varepsilon\, \Omega} F(\sigma, s)$. Then, the regret under scenario s for a sequence σ is denoted as (13).

$$R(\sigma, s) = F(\sigma, s) - F_s^* \tag{13}$$

Among all the possible scenarios, the one which maximises the regret of the schedule σ is called the worst-case scenario for sequence σ. The maximum regret of sequence σ is shown as (14).

$$R_{max}(\sigma, s) = \max_{s \in S}(F(\sigma, s) - F_s^*) \tag{14}$$

The proposed solution method aims to find a robust sequence σ with the minimal–maximum regret. It can be formulated as equation (15).

$$\min_{\sigma\in\Omega} R_{max}(\sigma, s) = \min_{\sigma\in\Omega}\left(\max_{s\in S}(F(\sigma, s) - F_s^*)\right) \tag{15}$$

A matrix $Y = [y_{hk}]_{h\times h}$ is used product sequencing in the considered two-stage production system. For This matrix, y_{hk} equals 1 if the product h is the kth product to be assembled; and 0 otherwise. For the processing stage, F_k^s is used to denote the assembling time of product occupying the kth position under the scenario s for the given schedule σ. Note that in any sequence, it is uncommon that a part i occupies a later position of processing than another part j related to a successor product.

According the above description, the proposed model entitled robust assembly flow shop (RAF) is presented as follows:

$$\min_{\sigma\in\Omega}\left(\max_{s\in S}(F(\sigma,s)-F_s^*)\right) \quad (16)$$

$$\sum_{h=1}^{H} y_{hk}=1 \qquad \forall k=1,2,\dots,H \quad (17)$$

$$\sum_{k=1}^{H} y_{hk}=1 \qquad \forall h=1,2,\dots,H \quad (18)$$

$$CP_{i1}\geq \sum_{h=1}^{H}(P_{ih}\times y_{h1}) \qquad \forall i \quad (19)$$

$$CP_{ik}\geq CP_{i(k-1)}+\sum_{h=1}^{H}(P_{ih}\times y_{hk}) \qquad \forall i,\forall k=2,3,\dots,H \quad (20)$$

$$F_k^s\geq CP_{ik} \qquad \forall i,k \quad (21)$$

$$F_k^s\geq F_{k-1}^s+A_h^s \qquad \forall k=2,3,\dots,H \quad (22)$$

$$y_{hk}\in\{0,1\} \qquad \forall k,h \quad (23)$$

Constraint (16) determine the value of the objective function. Constraints (17) and (18) guarantee that the robust sequence is feasible. Constraints (19) calculate the completion time of the parts related to the product that occupies the first position. Similarly, constraints (20) indicate the completion time of the parts related to the products that occupy the second to the last position. Constraints (21) and (2) calculate the Start time for of product assembly. Finally, constraints (23) demonstrate the change scope of decision variable.

In this model, F_s^* can be determined by solving a mixed-integer linear programming model. Since$F(\sigma,s)=\max_k\{F_k^s+A_k^s\}$, there are two nested max operators in the objective function. We transform it into a mixed-integer linear programming model as follows.

$$\min_{\sigma\in\Omega} r \quad (24)$$

$$F_k^s+A_k^s\text{-}r\leq F_s^* \qquad k=1,2,\dots,H,s\in S \quad (25)$$

(17) ~ (23)

In the above mixed-integer linear programming model, (17) ~ (23) means the equations (17), (18), (23) and constraints (19)-(21) of the previous model.

SOLUTION METHODS

I. Exact algorithm

In this section, a general iterative relaxation (IR) procedure is proposed to solve the min-max regret problem explained in the previous section. The considered procedure was introduced by [40] for the first time and developed by [41], [42]. This approach has been used in different studies due to its efficiency. For instance, we can cite [12] that used this method for a two-stage hybrid flow shop scheduling problem with uncertain interval processing times. Based on this method, the iteration can be reduced using a finite set Γ to replace all the possible scenarios S in the solution method. The set Γ is defined as follows:

$$\Gamma=\{A_h=A_h^{min}\ \text{ or }\ A_h^s=A_h^{max}\quad h=1,2,\dots,H\}$$

It is obvious that, $\Gamma\in S$ and the relaxed mixed-integer model called RAFS-relaxed in this paper can be presented as follows:

$$\min_{\sigma\in\Omega} r \quad (26)$$

$$F_k^s+A_k^s-r\leq F_{s_w}^* \quad k=1,2,\dots,H,\forall s_w\in\Gamma \quad (27)$$

(17) ~ (23)

The optimal makespan F_{s_w}can be provided by solving a mixed-integer programming for a determined scenario s_w. Let $\hat{\sigma}$denote the solution to RAFS-relaxed, with corresponding objective value $\hat{r}$. We use r^* to represent the minimal–maximum regret. Then, $\hat{r}\leq r^*$is a proper lower bound of the solution. The lower bound $\hat{r}$ is non-decreasing as more regret cuts being added.

We use a worst-case procedure to obtain the worst-case scenario and the corresponding maximum regret (R_{max}) for a specific sequence $\hat{\sigma}$. It is obvious that $R_{max}\geq r^*$is the upper bound of the solution. In the worst-case procedure, a state (σ,h) shows the sequence σ wherein the product his the critical product with the maximum makespan. In this way, for a given sequence $\hat{\sigma}$ we first calculate the deviation between the makespan of this sequence and the corresponding makespan of the optimal sequence.

During each iteration, the state $\hat{s}$ is chosen associated with the greatest deviation as the worst-case scenario of the sequence $\hat{\sigma}$. Then, by a candidate initial sequence $\hat{\sigma}$, we can get its worst-case procedure and then the upper bound. Add this scenario into Γ, and generate new regret cuts. Thus, H constraints, $F_k^s + A_k^s - r \le F_{s_w}^*$ as regret cuts are added into RAFS-relaxed.

Then, the new solution and the lower bound can be obtained by solving the new RAFS -relaxed. The iterative procedure continues until the upper bound is no greater than the current lower bound.

The problem at hand is well known strongly NP-hard and the relaxed mixed-integer program model is far more difficult since the number of regret cuts increases with the number of iterations. Besides, for each iteration, the upper bound $R_{max}(\hat{\sigma})$for a given sequence $\hat{\sigma}$ must be obtained by the worst-case procedure. However, for any given state (σ, h), we have to compute the deviation between the makespan of this sequence under this state and the corresponding makespan of the optimal sequence in the worst-case procedure. The NP-hard problem should to be solved n times in the worst-case procedure in each iteration. Therefore, with an inspiration from [12] we improve the considered IR algorithm by reducing the number of iterations. In this way, we always concern the least upper bound and corresponding sequence. Thus, we define UB^* and σ^* as the least upper bound and corresponding sequence respectively. Then the following steps are need to reduce iterations:

I. Input an initial sequence $\hat{\sigma}$. Set $LB = 0$, $UB^* = +\infty$ and $\sigma^* = \phi$;

II. Generate the worst-case $\hat{S}$ of $\hat{\sigma}$ by worst-case procedure and obtain the corresponding maximum regret $R_{max}(\hat{\sigma})$;
If $UB^* \ge R_{max}(\hat{\sigma})$ set $UB^* = R_{max}(\hat{\sigma})$ and $\sigma^* = \hat{\sigma}$ go to step 3
If $UB^* < LB$ go to step 5

III. Add the new regret cuts $F_k^{\hat{S}} + A_k^{\hat{S}} - r \le F_{\hat{S}}^*$ into RAFS-relaxed;

IV. Calculate $\hat{\sigma}$ and $\hat{r}$ by solving RAFS-relaxed then set $LB = \hat{r}$ and go step 2;

V. Stop and output the optimal sequence $\hat{\sigma}$r and UB^* as the minimal–maximum regret;

As mentioned before, the considered problem is a well-known NP-hard problem. Therefore, a heuristic solution method is proposed in the next section to solve the problem with practical scales in a reasonable time.

II. Approximate algorithms

Due to the complexity of the problem at hand, it is computationally expensive to provide the optimal solution using the exact method. Therefore, a heuristic and a metaheuristic algorithm are proposed in this section to find a near-optimal sequence for the products. The proposed heuristic solution method is developed based on the heuristic introduced by [39] that is modified for the uncertainty condition. According to this method, the solution procedure is divided into two phases. The first phase is determining the sequence of products to assemble and the second one is assigning the parts of each product h to a machine in the first stage to process (for $h = 1, 2, \ldots, H$).

For instance, suppose that three products are ordered to be produced ($H = 3$). The first product needs parts number 1, 2, and 3 to be assembled. The second product needs parts number 4, 5 and finally the third product needs parts number 6, 7 for assembling. Two aforementioned solution phases of this example can be demonstrated as Figure 2.

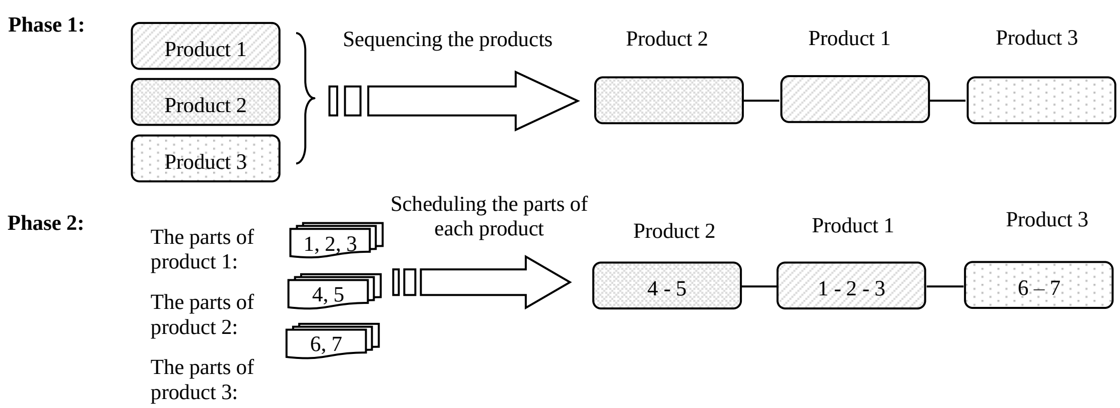

FIGURE 2
TWO-PHASE SOLUTION APPROACH

For the first phase (product sequencing) an extension of the Johnson's algorithm is used. After that, a heuristic method is proposed for assigning parallel machines to process the parts.

Johnson's rule that was introduced in 1954 is a method for job scheduling in the two-stage flow shop with the objective of minimizing Makespan (that is ($F_2 \parallel C_{max}$)). It also reduces the amount of idle time between the two stages in the optimal sequence. The technique requires several preconditions:

A summary of the Johnson's algorithm for scheduling jobs in the two-stage flow shop is given as below:

Suppose that p_{i1} is the processing time of job i in stage 1 and p_{i2} is the processing time of job i in stage 2. Similarly, p_{j1} and p_{j2} are processing time of job j in stage 1 and stage 2 respectively. In the optimal schedule, job i precedes job j if:

$$\min\{p_{i1}, p_{j2}\} < \min\{p_{i2}, p_{j1}\}$$

The steps of this algorithm can be presented as below:

- Step 1: Form the set U containing all jobs with $p_{i1} < p_{i2}$
- Step 2: Form the set V containing all jobs with $p_{i1} > p_{i2}$

  The jobs with $p_{i1} = p_{i2}$ can be put in either set.
- Step 3: Sort the jobs in set U in increasing order (SPT)
- Step 4: Sort the jobs in set V in decreasing order (LPT)
- Step 5: Form the sequence of all jobs according to the set U followed by the set V

To extend and use the Johnson's algorithm for the problem at hand, the parallel machines are assumed as the first stage of the production system and the assembly stage is considered as the second stage (or the first and the second machine respectively). Then the Johnson's method is used to determine the sequence of the products as the first phase. Assembly time of each product h is considered as p_{2h}. Also, the maximum time of p_{ih} ($i \in a, b$) can be computed as p_{1h}.

After determining the first sequence, we can calculate its worst-case scenario and then indicate the regret under this scenario. It should be noted that, constructing the initial sequence and doing the worst-case procedure require to provide the optimal sequence of the problem at hand. Since the problem is NP-hard, near-optimal solutions obtained by some approximate methods proposed by [5] and [39] are replaced.

By replacing the initial sequence with new one, new regret can be calculated and the maximum regret can be improved to get the near-robust sequence. There are two methods to change a sequence into a new one (i.e. shift neighborhood and interchange neighborhood). Shift neighborhood is the method moving product from one position to another, while interchange neighborhood is interchanging products between different positions. Jozefowska et al. [43][43]proved interchange neighborhood yields better results than the shift neighborhood by computational experimentation [43]. We also use interchange neighborhood for the considered problem to change the current sequence in each iteration.

In addition to the above mentioned exact and heuristic algorithms that are called EA and HA respectively in this study, a Hill-climbing local search algorithm (HLSA) proposed by [12] for scheduling of a two-stage hybrid flow shop with uncertain interval processing times is modified and used. Therefore, three algorithms EA, HA, and HLSA are applied to solve the problem and for result comparison.

COMPUTATIONAL EXPERIMENTATION

To use the proposed algorithms for solving the problem at hand, first, some test instances are design considering problem features. In this way, we follow the approach in [5] that has designed proper test examples for an assembly flexible flow shop scheduling problem with deterministic processing times. They designed 60 test examples under different various conditions. These examples are summarized for the considered problem in this study to one processing stage consists of several parallel machines and one assembly stage. Moreover, the number of products are considered 5, 10, and 15 and all characteristics of the test instances have been demonstrated in Table 1.

To modify the instances to features of the considered problem in this research under uncertainty, definitions of [12] are modified for the problem at hand in this paper. To this end, the lower bound of the assembly time for the product h is chosen at random from two uniform distributions of integers on the interval $\mathrm{A}_{\mathrm{h}}^{\min} \in [10\beta, (10+40\alpha_1)\beta]$. The upper bounds of the assembly time for the product h is chosen at random from two uniform distributions of integers on the interval $\mathrm{A}_{\mathrm{h}}^{\max} \in \left[A_h^{min}, (10+\alpha_2)A_h^{min}\right]$. Two parameters α_1 and α_2 are required to control the variability of assembly time across products and within a given product, respectively. Three values 0.2, 0.6, and 1.0 are used for both α_1 and α_2. Finally, the processing time of parts in the first stage is chosen at random from the uniform distribution [5 , 10]. We defined a range of processing time to evaluate the performance of the algorithms in different conditions.

TABLE 1
CHARACTERISTICS OF TEST INSTANCES.

Example No.	1	2	3	4	5	6	7	8	9
H	5	5	5	5	5	5	5	5	5
(α_1, α_2)	(0.2,0.2)	(0.2,0.6)	(0.2,1.0)	(0.6,0.2)	(0.6,0.6)	(0.6,1.0)	(1.0,0.2)	(1.0,0.6)	(1.0,1.0)
P_{ah}	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]
P_{bh}	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]

Table 1 (Continue)

Example No.	10	11	12	13	14	15	16	17	18
H	10	10	10	10	10	10	10	10	10
(α_1, α_2)	(0.2,0.2)	(0.2,0.6)	(0.2,1.0)	(0.6,0.2)	(0.6,0.6)	(0.6,1.0)	(1.0,0.2)	(1.0,0.6)	(1.0,1.0)
P_{ah}	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]
P_{bh}	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]

Table 1 (Continue)

Example No.	19	20	21	22	23	24	25	26	27
H	15	15	15	15	15	15	15	15	15
(α_1, α_2)	(0.2,0.2)	(0.2,0.6)	(0.2,1.0)	(0.6,0.2)	(0.6,0.6)	(0.6,1.0)	(1.0,0.2)	(1.0,0.6)	(1.0,1.0)
P_{ah}	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]
P_{bh}	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]	[5,10]

To modify the instances to features of the considered problem in this research under uncertainty, definitions of [12] are modified for the problem at hand in this paper. To this end, the lower bound of the assembly time for the product h is chosen at random from two uniform distributions of integers on the interval $A_h^{min} \in [10\beta\ , (10 + 40\alpha_1)\beta]$. The upper bounds of the assembly time for the product h is chosen at random from two uniform distributions of integers on the interval $A_h^{max} \in \left[A_h^{min}\ , (10 + \alpha_2)A_h^{min}\right]$. Two parameters α_1 and α_2 are required to control the variability of assembly time across products and within a given product, respectively. Three values 0.2, 0.6, and 1.0 are used for both α_1 and α_2. Finally, the processing time of parts in the first stage is chosen at random from the uniform distribution [5 , 10]. We defined a range of processing time to evaluate the performance of the algorithms in different conditions.

There are 10 replications generated for each combination of h, α_1 and α_2 resulting in 270 test instances and each test instance is solved by the three given algorithms. All procedures have been coded in MATLAB 2013 and the Computational experiments are executed on a Pc with a 2.0GHz Intel Core 2 Duo processor and 1GB of RAM memory.

Computational result and performance of the proposed algorithms in solving test instances has been demonstrated in Table 2. Two sub columns of the EA demonstrate the average iterations required to obtain the robust optimal solutions and the average CPU time (in seconds) respectively. For two approximate algorithms we have three sub columns that show the average CPU time (in seconds), the average percentage increase over the optimal sequence under the mid-point scenario, and the maximum percentage increase over the optimal sequence under the mid-point scenario respectively. Definition of midpoint scenario has been inspiration from the Xu et al. [33]. They introduced a scenario $S^{\frac{1}{2}}$ for the problem $P\|C_{max}$ in which the processing time of part j in stage i is determined as $p_{ij}^{S^{\frac{1}{2}}} = \frac{1}{2}\left(\underline{p}_{ij} + \bar{p}_{ij}\right)$. This definition is modified in this study for the considered problem as $A_h^{S^{\frac{1}{2}}} = \frac{1}{2}\left(A_h^{min} + A_h^{max}\right)$. They emphasized that a considerable initial sequence $\hat{\sigma}$ where jobs are scheduled by solving the problem $P\|C_{max}$ under the abovementioned scenario has the maximum regret $R_{max}(\hat{\sigma})$ with an upper bound. This property is |also applied to our problem.

The result shows that the CPU time of the exact algorithm increases rapidly when the problem size gets larger. However, the obtained exact solutions are considered as proper references for evaluation of the approximation methods. While the number of products grows to 15, the problem cannot be solved by the exact algorithm within 1 hour. Both of the proposed heuristic and metaheuristic algorithms consume less CPU time compared with the exact algorithm.

TABLE 2
COMPUTATIONAL RESULTS OF THE PROPOSED ALGORITHMS.

Example No.	*EA*		*HA*			*HLSA*			Mid-point		Makespan	
	Iteration	CPU	CPU	Mean	Max	CPU	Mean	Max	Mean	Max	Mean	Max
1	7.6	201.9	0.81	3.8	12.1	10.61	1.6	8	10.8	35.4	0.35	1.04
2	8.4	216.1	0.84	5.7	17.8	11.64	1.8	11.4	15.1	64.8	0.71	0.98
3	7.2	198.8	0.49	6.3	16.9	11.29	3.9	11.7	10.4	51.9	0.69	1.26
4	6.9	174.8	0.72	8.7	12.5	11.52	7.5	12.3	12.4	50.7	1.03	2.31
5	8.1	218.7	0.78	9.4	19.3	12.2	5.5	18.9	19.8	102.5	0.27	1.08
6	8.8	232.9	0.41	9.6	16.9	12.7	6.8	11.1	15.8	79	0.58	2.12
7	7.9	211.1	0.66	8.5	20.5	12.6	7.5	16.9	19.8	132.7	1.21	3.17
8	9.6	237.7	0.68	10.2	23.6	9.9	9.5	19	17.9	60.6	1.07	2.84
9	7.4	199.5	0.79	7.4	19.5	11.5	7.3	17.3	14.2	89.6	0.41	3.01
10	34.8	975.7	2.51	14.8	66.7	32.5	10.7	65.9	38.7	122.4	2.09	4.32
11	37.2	988.5	2.54	12.8	54.9	34.9	10.6	59.6	36.8	203.7	2.36	3.98
12	40.4	1062.1	2.19	16.2	67.4	42.4	7.3	67.3	40.6	231.7	1.98	5.27
13	38.6	1047.5	2.42	15.6	49.2	41	7.1	51.2	35.4	89.6	3.05	4.38
14	37.7.	1014.9	2.48	11.8	58.5	39.3	8.4	56.1	30.4	117.5	4.17	4.49
15	41.2	1149.3	2.11	14.3	54.4	41.9	9.1	44.6	36.8	240.3	2.75	5.94
16	40.3	1021.4	2.36	16.9	87.2	45.8	12.9	81.6	29.6	187.9	5.12	6.53
17	38.7	1035.7	2.38	17.1	49.8	33.7	10.2	74.6	24.6	154.6	3.66	4.05
18	39.4	1042.8	2.49	16.4	50.7	34.8	11.9	63.6	36.6	143.8	4.29	5.48
19	121.7	3406.2	9.7	21.7	72.8	96.3	11.8	91.8	47	212.7	6.41	9.09
20	128.6	3452.8	8.4	19.4	84.1	103.8	10.7	45.1	50.2	257.5	3.78	8.71
21	125.7	3304.9	7.8	19.9	79.4	107.3	12.5	51.8	44.8	195.9	5.16	12.02
22	130.8	3549.7	9.2	20.5	90.34	111.8	11.8	47.2	64.6	304.2	7.87	11.48
23	126.5	3448.2	10.4	23.6	88.4	114.2	14.4	94.8	66.9	278.9	6.09	12.56
24	122.9	3411.6	7.8	22.8	69.8	107.3	15	104.7	55.6	317.7	6.61	13.04
25	131.7	3559.3	9.8	20.2	102.6	122.9	14.8	83.5	40.7	121.7	5.9	10.89
26	130.8	3508.7	8.7	21.3	112.8	99.3	10.8	119.7	54.2	319.6	8.44	14.65
27	127.3	3451.2	8.6	19.7	96.2	98.8	11.3	103.8	52.4	270.3	7.65	13.27

To better comparison the CPU time, we can see the CPU time of the heuristics algorithm and Hill-climbing local search algorithm for solving the problem on different sizes in Figure 3. In this way, we continued to design the test examples for 25, 50, 75, and 100 products. As the figure shows, the CPU time of the Hill-climbing local search algorithm increases dramatically as the dimensions of the problem increase. However, the increasing CPU time of the heuristic algorithm increases at a very low rate.

It can be also concluded from this table that both two proposed algorithms closely approximate the optimal solution. However, HLSA obtains more close solutions to the optimal solution provided by the exact method. In addition, since HLSA has carried out more iterations and exploration, it costs more CPU times compared with HA.

Table 2 shows that the solutions quality of the optimal schedule under the mid-point scenario has significant deviations from the optimal condition. The average approximation of the result is from 10.8% to 66.9%, and the maximum errors of the method are between 35.4% and 319.6%. In addition, both the average and the maximum deviations rise as the problem size increases.

Moreover, the result denotes that the proposed robust schedule procedure closely approximates the schedule in terms of the optimal expected makespan. The average approximation errors of the robust scheduling is between 0.27% to 8.44%, and the maximum errors of 0.98% and 14.56%. We can conclude that the robust optimal schedules hedge against assembling time uncertainty while maintaining excellent expected makespan performance.

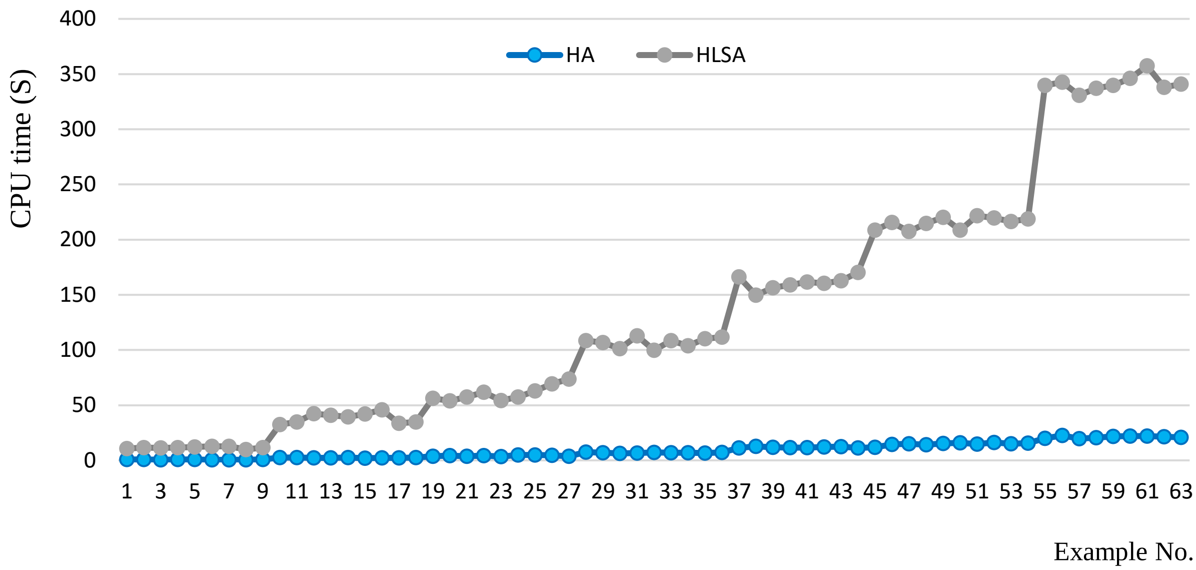

FIGURE 3
COMPARISON OF CPU TIMES

CONCLUSIONS

This paper focused on a two-stage assembly flow shop in which the assembling times of products are uncertain and the performance measure of interest is system makespan. The problem was described carefully and an MIP model was proposed for the deterministic environment. In addition, the problem was extended under uncertainty in assembling times of products. An iterative relaxation (IR) method was proposed as an exact algorithm to solve the problem on the small scales. In addition, one heuristic method and one metaheuristic (i.e. Hill-climbing local search algorithm (HLSA)) were tuned to solve the problem at hand on the medium and large sizes.

The considered problem has many application in manufacturing industries. Considering new features in this study especially investigating the problem under uncertainty in assembling times, close the problem to real-world condition. For the shop floor activities managers, it is vital to consider uncertainties in times of different operations in addition to the others disruptions.

Several number of numerical instances taken from valid references were modified for the considered problem to evaluate the performance of the proposed solution procedure. The computational results demonstrated that the proposed robust scheduling procedures hedge effectively against uncertain assembling times, while maintaining excellent expected makespan performance. However, the proposed heuristic consumes less CPU time compared with the metaheuristic algorithm and the metaheuristic outperforms the heuristic in terms of solution quality.

In practice, the processing times of parts on each stage and the assembling times of products on the last stage are often uncertain due to the machine conditions, worker skill levels, or some other accidental factors. Due to the result of this study, ignoring these facts may lead to lose the ideal solution that causes increasing in makespan and operational costs.

According to the finding of this study, the quality of the solution of the optimal schedule without considering changes in assembling times causes significant deviations from the optimal condition. The final result denoted that the average approximation of the result is from 10.8% to 66.9%, and the maximum errors of the method is between 35.4% and 319.6%. In addition, both the average and the maximum deviations rise as the problem size increases.

Especially, for the scheduling problems with delivery date and tardiness penalty, it is vital to determine appropriate time window according to the uncertainty in parameters. In addition, identifying the root of uncertainty factors and reducing their effects as much as possible can help to have a robust optimal schedule that significantly hedges against processing times and assembling times uncertainty.

It would be worth investigating situations where there is uncertainty in setup times. Other possible extensions of this work include considering the time window for the due date and investigating lateness as a new objective function. Extension of the proposed methods for other scheduling problems such as three-stage assembly flow shop or flow shop with assembly stage can be another attractive topic for future research.

References

[1] I. Hamdi and S. Toumi, "MILP models and valid inequalities for the two-machine permutation flowshop scheduling problem with minimal time lags," J. Ind. Eng. Int., vol. 15, no. 1, pp. 223–229, Dec. 2019, doi: 10.1007/S40092-019-00331-1.

[2] R. Buddala and S. Mahapatra, S., "Improved teaching–learning-based and JAYA optimization algorithms for solving flexible flow shop scheduling problems," J. Ind. Eng. Int., vol. 14, no. 3, pp. 555–570, Sep. 2018, doi: 10.1007/S40092-017-0244-4.

[3] M. Afzalirad and J. Rezaeian, "Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions," Comput. Ind. Eng., vol. 98, pp. 40–52, Aug. 2016, doi: 10.1016/J.CIE.2016.05.020.

[4] F. Daneshamooz, P. Fattahi, and S. M. H. Hosseini, "Scheduling in a flexible job shop followed by some parallel assembly stations considering lot streaming," Eng. Optim., pp. 1–20, 2021, doi: 10.1080/0305215X.2021.1887168.

[5] P. Fattahi, S. M. H. Hosseini, and F. Jolai, "A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem," Int. J. Adv. Manuf. Technol., vol. 65, no. 5–8, pp. 787–802, 2013, Accessed: Aug. 07, 2021. [Online]. Available: https://doi.org/10.1007/s00170-012-4217-x.

[6] T. Chaari, S. Chaabane, T. Loukil, and D. Trentesaux, "A genetic algorithm for robust hybrid flow shop scheduling," Int. J. Comput. Integr. Manuf., vol. 24, no. 9, pp. 821–833, 2011, doi: 10.1080/0951192X.2011.575181.

[7] R. Montemanni, "A mixed integer programming formulation for the total flow time single machine robust scheduling problem with interval data," J. Math. Model. Algorithms, vol. 6, no. 2, pp. 287–296, Jun. 2007, doi: 10.1007/S10852-006-9044-3.

[8] A. Allahverdi and H. Aydilek, "Heuristics for the two-machine flowshop scheduling problem to minimise makespan with bounded processing times," Int. J. Prod. Res., vol. 48, no. 21, pp. 6367–6385, Nov. 2010, doi: 10.1080/00207540903321657.

[9] Z. Ziaei and A. Jabbarzadeh, "A multi-objective robust optimization approach for green location-routing planning of multi-modal transportation systems under uncertainty," J. Clean. Prod., vol. 291, no. 0959–6526, p. 125293, 2021, Accessed: Aug. 20, 2021. [Online]. Available: https://doi.org/10.1016/j.jclepro.2020.125293.

[10] N. M. Matsveichuk, Y. Sotskov, N. G. Egorova, and T. C. Lai, "Schedule execution for two-machine flow-shop with interval processing times," Math. Comput. Model., vol. 49, no. 5–6, pp. 991–1011, 2009, [Online]. Available: doi.org/10.1016/j.mcm.2008.02.004.

[11] N. M. Matsveichuk, Y. N. Sotskov, and F. Werner, "The dominance digraph as a solution to the two-machine flow-shop problem with interval processing times," Optimization, vol. 60, no. 12, pp. 1493–1517, Dec. 2011, doi: 10.1080/02331931003657691.

[12] X. Feng, F. Zheng, and Y. Xu, "Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times," Int. J. Prod. Res., vol. 54, no. 12, pp. 3706–3717, Jun. 2016, doi: 10.1080/00207543.2016.1162341.

[13] G. M. Komaki, E. Teymourian, V. Kayvanfar, and Z. Booyavi, "Improved discrete cuckoo optimization algorithm for the three-stage assembly flowshop scheduling problem," Comput. Ind. Eng., vol. 105, pp. 158–173, 2017, Accessed: Aug. 07, 2021. [Online]. Available: https://doi.org/10.1016/j.cie.2017.01.006.

[14] A. Allahverdi and F. S. Al-Anzi, "The two-stage assembly scheduling problem to minimize total completion time with setup times," Comput. Oper. Res., vol. 36, pp. 2740–2747, 2009, doi: 10.1016/j.cor.2008.12.001.

[15] C. Y. Lee, T. C. E. Cheng, and B. M. T. Lin, "Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem," Manage. Sci., vol. 39, no. 5, pp. 616–625, 1993, doi: 10.1287/MNSC.39.5.616.

[16] C. N. Potts, S. V. Sevast'janov, V. A. Strusevich, L. N. Van Wassenhove, and C. M. Zwaneveld, "The two-stage assembly scheduling problem: Complexity and approximation," Comput. Oper. Res., vol. 43, no. 2, pp. 346–355, Apr. 1995, doi: 10.1287/OPRE.43.2.346.

[17] X. Sun, K. Morizawa, and H. Nagasawa, "Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling," Eur. J. Oper. Res., vol. 146, no. 3, pp. 498–516, 2003, Accessed: Aug. 07, 2021. [Online]. Available: https://doi.org/10.1016/S0377-2217(02)00245-X.

[18] C. Koulamas and G. j. Kyparisis, "The three-stage assembly flowshop scheduling problem," Comput. Oper. Res., vol. 28, no. 7, pp. 689–704, 2001, Accessed: Aug. 22, 2021. [Online]. Available: https://doi.org/10.1016/S0305-0548(00)00004-6.

[19] A. Maleki-Darounkolaei, M. Modiri, R. Tavakkoli-Moghaddam, and I. Seyyedi, "A three-stage assembly flow shop scheduling problem with blocking and sequence-dependent set up times," J. Ind. Eng. Int., vol. 8, no. 1, pp. 1–7, Dec. 2012, doi: 10.1186/2251-712X-8-26.

[20] P. Fattahi, S. M. H. Hosseini, and F. Jolai, "A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations," Appl. Math. Model., vol. 38, no. 1, pp. 119–134, 2014, Accessed: Aug. 08, 2021. [Online]. Available: https://doi.org/10.1016/j.apm.2013.06.005.

[21] J. M. Framinan and p. p. Gonzalez, "The 2-stage assembly flowshop scheduling problem with total completion time: Efficient constructive heuristic and metaheuristic," Comput. Oper. Res., vol. 88, no. 0305–0548, pp. 237–246, 2017, doi: 10.1016/j.cor.2017.07.012.

[22] G. M. Komaki, S. Sheikh, and B. Malakooti, "Flow shop scheduling problems with assembly operations: a review and new trends," Int. J. Prod. Res., vol. 57, no. 10, pp. 2926–2955, May 2019, doi: 10.1080/00207543.2018.1550269.

[23] C. C. Wu et al., "A branch-and-bound algorithm and four metaheuristics for minimizing total completion time for a two-stage assembly flow-shop scheduling problem with learning consideration," Eng. Optim., vol. 52, no. 6, pp. 1009–1036, Jun. 2020, doi: 10.1080/0305215X.2019.1632303.

[24] J. Deng, L. Wang, S. Y. Wang, and X. L. Zheng, "A competitive memetic algorithm for the distributed two-stage assembly flow-shop scheduling problem," Int. J. Prod. Res., vol. 54, no. 12, pp. 3561–3577, Jun. 2016, doi: 10.1080/00207543.2015.1084063.

[25] D. Lei, B. Su, and M. Li, "Cooperated teaching-learning-based optimisation for distributed two-stage assembly flow shop scheduling," Int. J. Prod. Res., pp. 1–14, 2020, doi: 10.1080/00207543.2020.1836422.

[26]J. Huang and X. Gu, "Distributed assembly permutation flow-shop scheduling problem with sequence-dependent set-up times using a novel biogeography-based optimization algorithm," Eng. Optim., 2021, doi: 10.1080/0305215X.2021.1886289.

[27]P. Kouvelis, R. L. Daniels, and G. Vairaktarakis, "Robust scheduling of a two-machine flow shop with uncertain processing times," IIE Trans., vol. 32, no. 5, pp. 421–432, 2000, doi: 10.1023/A:1007640726040.

[28]M. T. Jensen, "Neighbourhood based robustness applied to tardiness and total flowtime job shops," Parallel Probl. Solving from Nat. PPSN VI. PPSN 2000. Lect. Notes Comput. Sci., vol. 1917, pp. 283–292, 2000, doi: 10.1007/3-540-45356-3_28.

[29]J. C. Billaut, A. Moukrim, and E. Sanlaville, Flexibility and robustness in scheduling. John Wiley, 2008.

[30]J. Balasubramanian and I. E. Grossmann, "A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times," Comput. Chem. Eng., vol. 26, no. 1, pp. 41–57, 2002, Accessed: Aug. 23, 2021. [Online]. Available: https://doi.org/10.1016/S0098-1354(01)00735-9.

[31]Z. Li and M. Ierapetritou, "Process scheduling under uncertainty: Review and challenges," Comput. Chem. Eng., vol. 32, no. 4–5, pp. 715–727, 2008, Accessed: Aug. 07, 2021. [Online]. Available: https://doi.org/10.1016/j.compchemeng.2007.03.001.

[32]P. M. Verderame, J. A. Elia, J. Li, and C. A. Floudas, "Planning and scheduling under uncertainty: A review across multiple sectors," Ind. Eng. Chem. Res., vol. 49, no. 9, pp. 3993–4017, May 2010, doi: 10.1021/IE902009K.

[33]X. Xu, W. Cui, J. Lin, and Y. Qian, "Robust makespan minimisation in identical parallel machine scheduling problem with interval data," Int. J. Prod. Res., vol. 51, no. 12, pp. 3532–3548, Jun. 2013, doi: 10.1080/00207543.2012.751510.

[34]M. González-Neira, E., R. Montoya-Torres, j., and D. Barrera, "Flow-shop scheduling problem under uncertainties: Review and trends," Int. J. Ind. Eng. Comput., vol. 8, no. 4, pp. 399–426, 2017.

[35]S. Tadayonirad, H. Seidgar, H. Fazlollahtabar, and R. Shafaei, "Robust scheduling in two-stage assembly flow shop problem with random machine breakdowns: integrated meta-heuristic algorithms and simulation approach," Assem. Autom., vol. 39, no. 5, pp. 944–962, Oct. 2019, doi: 10.1108/AA-10-2018-0165/FULL/HTML.

[36]M. F. Amiri and J. Behnamian, "Multi-objective green flowshop scheduling problem under uncertainty: Estimation of distribution algorithm," J. Clean. Prod., vol. 251, 2020, Accessed: Aug. 07, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652619346049.

[37]C.-C. Wu, J. N. D. Gupta, S. R. Cheng, B. M. T. Lin, S. H. Yip, and W. C. Lin, "Robust scheduling for a two-stage assembly shop with scenario-dependent processing times," Int. J. Prod. Res., vol. 59, no. 17, pp. 5372–5387, 2021, doi: 10.1080/00207543.2020.1778208.

[38]P. Zheng, P. Zhang, J. Wang, J. Zhang, C. Yang, and Y. Jin, "A data-driven robust optimization method for the assembly job-shop scheduling problem under uncertainty," Int. J. Comput. Integr. Manuf., pp. 1–16, 2020, doi: 10.1080/0951192X.2020.1803506.

[39]S. M. H. Hosseini, "Modelling and solving the job shop scheduling Problem followed by an assembly stage considering maintenance operations and access restrictions to machines," J. Optim. Ind. Eng., vol. 12, no. 1, pp. 63–78, 2019, doi: 10.22094/joie.2018.760.1484.

[40]M. Inuiguchi and M. Sakawa, "Minimax regret solution to linear programming problems with an interval objective function," Eur. J. Oper. Res., vol. 86, no. 3, pp. 526–536, 1995, Accessed: Aug. 08, 2021. [Online]. Available: https://doi.org/10.1016/0377-2217(94)00092-Q.

[41]H. E. Mausser and M. Laguna, "A new mixed integer formulation for the maximum regret problem," Int. Trans. Oper. Res., vol. 5, no. 5, pp. 389–403, 1998, doi: 10.1111/J.1475-3995.1998.TB00122.X.

[42]H. E. Mausser and M. Laguna, "A heuristic to minimax absolute regret for linear programs with interval objective function coefficients," Eur. J. Oper. Res., vol. 117, no. 1, pp. 157–174, 1999, Accessed: Aug. 07, 2021. [Online]. Available: https://doi.org/10.1016/S0377-2217(98)00118-0.

[43]J. Jozefowska, M. Mika, R. Rozycki, G. Waligora, and J. Weglarz, "Local search metaheuristics for discrete–continuous scheduling problems," Eur. J. Oper. Res., vol. 107, no. 2, pp. 354–370, 1998, Accessed: Aug. 07, 2021. [Online]. Available: https://doi.org/10.1016/S0377-2217(97)00345-7.