Journal of Industrial and Systems Engineering
Vol. 14, No. 2, pp. 245-267
Spring (April) 2022

Investigating the two-stage assembly flow shop scheduling problem with uncertain assembling times

Maryam Seyedhamzeh[1], Hossein Amoozadkhalili[2*], Seyed Mohammad Hassan Hosseini[3], Morteza Honarmand Azimi[4], Kamaladdin Rahmani[4]

[1]*Department of Industrial Management, Tabriz Branch, Islamic Azad University, Tabriz, Iran*
[2]*Department of Industrial Engineering, Sari Branch, Islamic Azad University, Sari, 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*

seyedhamzeh_m@yahoo.com, amoozad92@yahoo.com, sh.hosseini@shahroodut.ac.ir, honarmand@iaut.ac.ir, kr13452000@yahoo.com

Abstract

The majority of scheduling research considers a deterministic environment with pre-known and fixed data. However, under the tools conditions and worker skill levels in assembly work stations, there is uncertainty in the assembling times of the products. This study aims to address a two-stage assembly flow shop scheduling problem with uncertain assembling times of the products which is assumed to follow a normal distribution. The problem is formulated as an MIP model in general form and under deterministic condition. Since the problem is strongly NP-hard, genetic algorithm is adopted with a new solution structure and fitness function to solve the problem on the practical scales. The presented robust procedure aims to maximize the probability of ensuring that makespan will not exceed the expected completion time. In addition, Johnson’s rule is extended and simulated annealing algorithm is tuned for the problem at hand. The computational results indicate that the obtained robust schedules hedge effectively against uncertain assembling times. The results also show that the proposed genetic algorithm gets better robust schedules than Johnson’s rule and outperforms simulated annealing algorithm in terms of deviation percentage ($\%D$) of the expected makespan from the optimal schedule.
Keyword: Scheduling, two-stage assembly flow shop, uncertainty, robustness, genetic algorithm

1-Introduction

A two-stage assembly flow shop is a special manufacturing environment, which consists of a fabrication stage followed by an assembly stage. Required components are processed at the first stage which consists of several parallel machines (Kazemi et al., 2017). After preparing a set of required components, they are assembled into the final product in the assembly stage. This type of production system has many applications especially in manufacturing industries wherein, complex products are produced through a combination of processing and assembly structures.

*Corresponding author
ISSN: 1735-8272,

For instance, we can point out fire engine assembly plant (Lee et al., 1993), database distribution (Allahverdi & Al-Anzi, 2006), and bedroom furniture manufacturing (Navaei et al., 2013).

The majority of the methods have been proposed under the traditional assumptions especially that the data are perfectly known and fixed. However, in real-world condition, different kinds of uncertain events may occur in processing an assembling operation. These events make especially the assembling time of products uncertain, which lead to makespan variations. These makespan variations cause the products to accomplish after the expected completion time. When makespan exceeds the expected completion time of products, it may alter the delivery schedule of products and leads to late delivery of products and increasing operational costs.

Whenever processing and assembling time can be modeled as a random variable, we can tackle the uncertainty using stochastic approaches. A proper scheduling system for a stochastic environment should not only be able to handle the uncertain events once they have happened but also be able to generate schedules that are prepared for these events. To this end, historical experiences and data are useful to obtain the probability distribution of uncertain factors. In some other cases, scenario-based approaches are 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 concerning the scheduling decision variables (Ziaei & Jabbarzadeh, 2021).

One important way to deal with uncertainty in scheduling is robust scheduling. Li et al. introduced robust scheduling as a schedule whose performance does not significantly degrade in the face of disruption (Li & Ierapetritou, 2008). Therefore, robust scheduling is currently used in a variety of applications where the aim is to control the performance degradation of schedules due to uncertainties. Some of the practical applications of robust scheduling are in project scheduling (Herroelen & Leus, 2004), manufacturing industries (Tang & Wang, 2008), and airline crew scheduling (Lan et al., 2006). Robust scheduling approaches in the existing literature can be categorized into reactive scheduling and proactive scheduling (Ghezail et al., 2010; Liu et al., 2011). Reactive scheduling techniques include all methods of robust scheduling that do not directly consider the uncertainty in generating schedules. These models tackle the uncertainty issue with revising the schedule when unexpected events occur (Sabuncuoglu & Goren, 2009). Proactive scheduling techniques that most of the current research is focused on, take potential disruptions into consideration during the generation of the initial schedule (Goren & Sabuncuoglu, 2008).

Given the importance of tackling the issue of uncertainty in assembly-type scheduling problems, this study aims to address a two-stage assembly flow shop scheduling problem wherein, the assembling times of products are uncertain in the assembly stage. Therefore, after definition the problem with all considered features, an improved genetic algorithm is proposed to get the robust schedule for the problem at hand. The main idea of the proposed approach is to maximize the probability of ensuring that makespan will not exceed the expected completion time.

The outline of the paper is as follows. After presenting an introduction to the considered problem in section 1, section 2 is devoted to the survey of studies related to this work. The problem is described and formulated in section 3. In Section 4, we present solution approaches including an improved genetic algorithm, simulated annealing, and a procedure based on the main idea of Johnson's rule. Result analysis has been provided in section 5. Finally, the conclusion and some suggestions for future studies are discussed in section 6.

2-Literature review

The literature related to this work is categorized into two subsections. First, the studies dealing with the two-stage assembly flow shop scheduling problem are presented. After that, the efforts focusing on the uncertainty issue in processing or assembling time of scheduling problems are investigated. Finally, the research gap is discussed to clear this paper's novelty.

2-1-Literature review on the two-stage assembly flow shop scheduling problem

Two-stage production environments that consider both the processing and assembly operation concurrently are one of the most popular scheduling problems in manufacturing industries. The two-stage assembly flow shop scheduling problem (TAFP) as an especial form of two-stage production systems was introduced by Lee et al. for the first time in 1993 (Lee et al., 1993). After that, many researchers deal with this problem considering makespan as the dominating objective function; for instance, we can cite (Sung & Kim, 2008), (Sung & Juhn, 2009), and (Navaei et al., 2013).

Some new research have investigated the problem considering other objective functions. For example, Allahverdi and Al-Anzi tackle this problem to minimize makespan and mean completion time of products simultaneously using a bi-objective model. The authors proposed three algorithms including simulated annealing (SA), ant colony optimization (ACO), and self-adaptive differential evolution (SDE) for the problem (Allahverdi & Al-Anzi, 2008). Torabzadeh and Zandieh dealt with the TAFP considering the same objective functions as (Allahverdi & Al-Anzi, 2008) and used the cloud theory-based simulated annealing (CSA) algorithm which was indicated to perform better than the SA (Torabzadeh & Zandieh, 2010). Allahverdi and Aydilek investigated the TAFP to minimize total tardiness for the first time. They proposed an insertion algorithm, a genetic algorithm, two versions of the simulated annealing (SA) algorithm, and two versions of cloud theory-based SA to solve the problem (Allahverdi & Aydilek, 2015).

Some studies extended the TAFP to the three-stage assembly flow shop by considering a supplementary stage between the processing and assembly stage wherein, the parts are collected and prepared. Koulamas and Kyparisis considered an intermediate operation after the machining stage devoted to collecting and transporting the fabricated parts from the processing areas to the assembly area. They analyzed the worst-case ratio bound for several heuristics to the problem on the large scales (Koulamas & Kyparisis, 2001). Similarly, Komaki et al. investigated a three-stage assembly flow shop scheduling problem wherein, the first stage includes several identical parallel machines followed by the second and the third stages that each of them has 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 (Komaki et al., 2017). Some studies have dealt with two-stage production systems using exact methods just for special cases. We can refer to (Wu et al., 2020) and (Daneshamooz et al., 2021) who developed a branch and bound algorithm with some tight lower bounds for this problem.

Recently efforts have considered practical features in TAFP to close it to real-world condition. For example, Lei et al. proposed a cooperated teaching-learning-based optimisation (CTLBO) to minimise makespan in a distributed two-stage assembly flow shop (Lei et al., 2020). Zhang and Tang incorporated flexible preventive maintenance (PM) operation into a two-stage assembly flow shop with m dedicated machines in the first (fabrication) stage and one machine in the second (assembly) stage. They formulated the problem using an MIP model considering maintenance level constraints with the aim of minimising the total completion time and maintenance time (Zhang & Tang, 2021b). The authors performed a similar study by incorporating preventive maintenance (PM) operation into TAFP where there are m_1 dedicated machines in fabrication stage and m_2 machines in the assembly stage. The main idea in their study is to find a fit product sequence along with PM execution time points. They proposed two heuristics and a PM-based iterated greedy algorithm for the problem (Zhang & Tang, 2021a).

2-2-Literature review on scheduling problems under uncertainty

In the real case, however, the processing times of jobs on each stage are often uncertain due to the assembly machine conditions, tool accuracy, worker skill levels, and some other accidental factors (Allahverdi & Aydilek, 2010). These uncertain environments required proper solution approaches to tackle different kinds of uncertainty. Despite the notable effects of uncertainty on the obtained result for production planning and scheduling problems, this issue has received relatively little attention in the literature of scheduling. In the literature, we can see some efforts that have assumed independent and known processing time distributions for individual jobs and proposed stochastic methods (Kouvelis et al., 2000). Furthermore, some other studies have followed the 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 (Freeman et al., 2016). In this section, we present the related works that have tackled the uncertainty in scheduling problems. The term robust is often defined as a phrase describing a solution that does not change its performance much if uncertain parameters or unexpected events occur. Billaut et al. (2008) state that a schedule is robust if its performance is relatively insensitive to the data uncertainty. In this article, we consider the latter definition of robust scheduling.

Balasubramanian and Grossmann (2002) addressed a multi-period flow shop scheduling problem with uncertain processing times. They formulated the problem as an MIP model with the aim of minimising the expected makespan. In addition, the authors developed a branch and bound algorithm with an aggregated probability model to solve a special case of the problem. Li and Ierapetritou (2008) emphasized that uncertainty is a very important issue in scheduling since it can cause infeasibilities and production disturbances. Therefore, they performed a comprehensive review on the main methodologies that have tackled the issue of uncertainty in scheduling problems until 2007 as well as identified the main challenges in this area. Similarly, Verderameet al. (2010) provided an overview of the key contributions within the planning and scheduling communities with specific emphasis on uncertainty analysis until 2009.

Kasperski et al. (2012) discussed the two-stage permutation flow shop problem with uncertain job processing times. They assumed that processing times are specified as a discrete scenario set and applied the min–max and min–max regret criteria to tackle the problem. They also approved that the min–max and min–max regret versions of the problem are strongly NP-hard even for two scenarios. González-Neira et al. (2017) surveyed papers about flow shop and flexible flow shop scheduling problems under uncertainty published from 2001 to October 2016 and drew up interesting topic further research in this area. Zheng et al. (2020) investigated a scheduling problem in assembly manufacturing systems under uncertainty in processing time and random machine breakdown. They proposed some robust methods to minimise makespan and the deviation of the actual schedule from the baseline schedule simultaneously.

Liao and Fu (2020) studied the permutation flow shop scheduling problem with interval production time. They developed a min–max regret criterion-based robust method to minimize the total completion time and the tardiness of production simultaneously. Moreover, a genetic algorithm was adopted and implemented to solve this robust scheduling model on the large sizes. Recently, Wang et al. (2021) conducted the job-shop scheduling problem with uncertain processing times using a discrete scenarios approach. Their objective functions were to minimize the mean makespan and the worst-scenario makespan across all the scenarios. The authors proposed two hybrid algorithms by combining the elitist nondominated sorting genetic algorithm (NSGA-II) and Tabu search (TS) operators to solve the problem.

Literature review indicates that the two-stage production systems including a processing stage followed by an assembly stage are receiving increasing attention in the field of academic research and manufacturing enterprise. .In practice, however, the exact assembling times of products are often uncertain due to the machine conditions, worker skill levels, and some other accidental factors. Robust approach, which is often concerned by risk-averse decision-makers, is focused on hedging against the worst-case performance rather than optimizing expected performance under all potential scenarios. As is evident in the existing literature, there is a lack of solution procedures for the assembly flow shop under uncertainty. There are two kinds of methods in robust scheduling to describe the set of all the possible scenarios: discrete processing time scenarios and continuous processing time intervals. The latter is applied in this paper. In this way, we improve genetic algorithm to tackle the problem wherein, there is uncertainty in the assembling times of the products which are assumed to follow a normal distribution. Moreover, two other methods are tuned and implemented based on simulated annealing and Johnson's rule to evaluate the robustness of the obtained solutions.

3-Problem statement

The considered two-stage assembly flow shop scheduling problem in this study has $m \geq 2$ identical parallel machines at the first stage and one assembly station at the second stage. A number of products of different kinds are ordered to produce. Each product is assembled with a set of specific parts. All parts are processed and ready in the first stage by the parallel machines and then, they are assembled into the final product at the second stage. According to the conditions of the assembly machine, different worker skill levels, and operational factors, there is uncertainty in the assembling times of the products which are assumed to follow a Truncated Normal distribution. We consider minimizing the maximum completion time of all products (makespan) as the objective function for this study. Figure 1 demonstrates a schematic view of the considered problem in this study.

Despite the huge number of works devoted to flow shop scheduling problems, there are just a few studies concerning assembly flow shop versions of these problems. These few efforts have dealt with the assembly flow shop assuming deterministic processing times of components and assembling times of products. The main assumptions of the considered problem are as follows:

- All parts are available for processing at time zero.
- There is no idle time on all processing machines at the first stage.
- Setup times are included in the processing time of parts.
- The processing machines are always available all times.
- Transportation time between the two stages is negligible
- All machines can process all kinds of parts in the processing stage but each machine can process only one part at the same time
- Each part can process on a machine at once.
- Parts can wait in an unlimited buffer space between two stages.
- The assembling operation of every product starts after preparing all the relevant parts and if the assembly stage is idle.
- The assembling times of the products are uncertain and follow a Truncated Normal distribution.

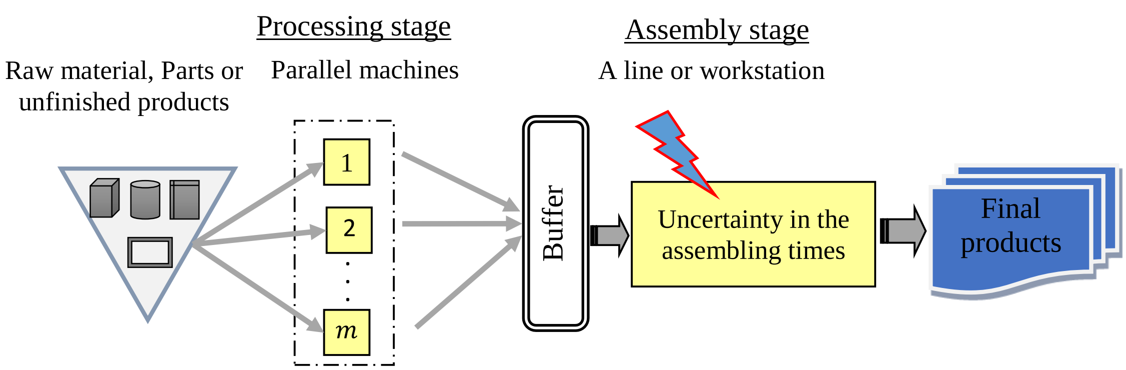

Fig 1. A schematic view of the considered problem

To clarify the problem at hand, we first formulate it under deterministic condition by proposing an MIP and then, the problem will be discussed under uncertainty in assembling times. To this end, first, the complete notation and the mathematical model are presented as below.

3-1-Sets and indices

$h, h' = \{1, 2, \dots, H\}$	Indices for products
$j, j' = \{1, 2, \dots, n\}$	Indices for parts
$J_h \subset n$	Set of parts for product h
$l = \{1, 2\}$	Set of production stages
$i = \{1, 2, \dots, m\}$	Set of parallel machines at the first stage

3-2- Parameters

P_j	Processing time of part j at the first stage
A_h	Assembling time of product h
L	A large number

3-3-Decision variables

$AS_{h,h'}$	Binary variable taking value 1 if product h is an immediate predecessor of product h' and 0 otherwise
Y_{ij}	Binary variable taking value 1 if part$j$ is processed on machine i; 0, otherwise
$X_{ijj'}$	Binary variable taking value 1 if part j' is assigned to machine i immediately after part j ; 0, otherwise
F_h	Start time for assembly of product h
C_h	Completion time of assembling the product h
CP_j	Completion time of part j at the processing stage
C_{max}	Maximum completion time of all products that is a continuous positive variable

3-4-Mathematical model

$$\text{Min } Z = C_{max} \tag{1}$$

Subject to:

$$\sum_{i=1}^{m} Y_{ij} = 1 \qquad \forall j \tag{2}$$

$$\sum_{\substack{j=0 \\ (j \neq j')}}^{n} \sum_{i=1}^{m} X_{ijj'} \leq 1 \qquad \forall j' \tag{3}$$

$$\sum_{\substack{j=1 \\ (j \neq j')}}^{n} \left(X_{ijj'} + X_{ij'j}\right) \leq 2Y_{ij'} \qquad \forall i, j' \tag{4}$$

$$\sum_{\substack{j'=1 \\ (j' \neq j)}}^{n} \sum_{i=1}^{m} X_{ijj'} \leq 1 \qquad \forall j \tag{5}$$

$$\sum_{j'=1}^{n} X_{i0j'} = 1 \qquad \forall\, i \qquad (6)$$

$$\sum_{i=1}^{m} \left(X_{ijj'} + X_{ij'j}\right) \leq 1 \qquad \forall\, j = 1, 2, \dots, n-1;\ j' > j \qquad (7)$$

$$CP_j \geq P_j \qquad \forall j \qquad (8)$$

$$CP_j \geq CP_{j'} + P_j + \left(\sum_{i=1}^{m} X_{ij'j} - 1\right).L \qquad \forall\, j, j', ;\ j \neq j' \qquad (9)$$

$$\sum_{\substack{h=0 \\ (h \neq h')}}^{H} AS_{hh'} = 1 \qquad \forall\, h' \qquad (10)$$

$$\sum_{h'=1}^{H} AS_{0h'} = 1 \qquad (11)$$

$$\sum_{\substack{h'=1 \\ (h' \neq h)}}^{H} AS_{hh'} \leq 1 \qquad \forall\, h \qquad (12)$$

$$AS_{hh'} + AS_{h'h} \leq 1 \qquad \forall\, h = 1, 2, \dots, H-1;\ h' > h \qquad (13)$$

$$F_h \geq CP_j \qquad \forall h;\ j \in \{J_h\} \qquad (14)$$

$$F_h \geq C_{h'} + (AS_{h'h} - 1).L \qquad \forall\, h, h' (h \neq h') \qquad (15)$$

$$C_h \geq F_h + A_h \qquad \forall\, h \qquad (16)$$

$$C_{max} \geq C_h \qquad \forall\, h \qquad (17)$$

$$X_{ijj'}, Y_{ij},\ AS_{h'h}\ \in \{0,1\} \qquad \forall\, i, j, j', h, h'\ (j \neq j';\ h \neq h') \qquad (18)$$

$$CP_j, C_h\ , F_h\ , C_{max} \geq 0 \qquad \forall j, h \qquad (19)$$

In the mathematical model, equation (1) minimize the maximum completion time of products (makespan) as the objective function. Relation set (2) assures that every part must be exactly assigned to one machine. Relation set (3) enforces that every part must be exactly at one position and only by one machine. Constraints (4) state that every part can be either a successor or predecessor on the processing machine to which it is assigned. Constraints (5) guarantee that every part has at most one succeeding part whereas relation set (6) controls that dummy part 0 has exactly one successor on each machine. Constraints (7) avoid the occurrence of cross-precedence, meaning that a part cannot be at the same time both a predecessor and a successor of another part. Constraints (8) and (9) calculate the completion time of parts. It is necessary to point out that constraint set (9) enforces that a machine can process at most one part at a time. Constraints (10) – (13) guarantee sequencing rules related to products assembly. According to constraints (10) every product has at most one preceding product including the dummy product. We need constraint (11) to ensure that just one product is assembled as the first product. Constraints (12) indicate that every product has at most one succeeding product. Constraints (13) avoid the occurrence of cross-precedencies, meaning that a product cannot be at the same time both a predecessor and a successor of another product. Constraints (14) and (15) calculate the assembly start time for each product. Constraints (16) determine the completion time of each product. Constraints (17) defines the makespan. Finally, constraints (18) and (19) define the domain of the decision variables.

It should be pointed out that we introduced a dummy part 0 with zero processing time, which precedes the first part for processing on each parallel machine. Similarly, a dummy product was considered with zero assembling time that precedes the first product at the assembly stage.

For the majority of deterministic scheduling problems in the literature, the processing and assembling times are considered deterministic and constant. However, there is uncertainty in processing times and/or assembling times in various real-life systems due to the machine conditions, worker skill levels, or some other accidental factors (Allahverdi & Aydilek, 2010). These uncertain environments can often be solved by some stochastic models if the probability distribution of processing time is determined. Historical data and experience are useful to obtain the probability distribution.

As is evident in the existing literature, the probabilistic technique is significant to represent processing and assembling time uncertainties. The processing time uncertainty is described by its distribution. The distribution of processing and assembling time can be obtained by collecting and analyzing the actual data of job processing times in every shop scheduling such as two-stage assembly flow shops or by assumptions. To get the assembling time distribution, a large quantity of assembling time data of repeat production in a case study in the automotive manufacturing industry was considered. The result indicated that the uncertain assembling time of products is an independent random variable that follows a Truncated Normal distribution. Therefore, an objective to maximize the probability that the maximum completion time (makespan) will be less than the expected makespan is introduced as (20) for the problem at hand instead of (1) in the deterministic model.

$$\text{Max } [P(C_{max} \leq \text{expected makespan})] \tag{20}$$

Uncertain assembling time of products is an independent random variable and so, due to the central limit theorem, the completion time of all products will follow normal distribution. It is well-known that the sum of two or more independent normally distributed numbers is also normally distributed. This fact can be shown for two random variables A and B with normally distributed as bellow:

$$\begin{matrix} A \sim N(\mu_A , \sigma_A^2) \\ B \sim N(\mu_B , \sigma_B^2) \end{matrix} \quad \Rightarrow \quad A + B \sim N(\mu_A + \mu_B , \sigma_A^2 + \sigma_B^2)$$

For a two-stage assembly flow shop scheduling of H products where the assembling time of products are uncertain, the robust schedule is defined as a schedule that gives the maximum probability that the makespan of the schedule will not exceed the expected completion time X. Mathematically, Probability $(\text{makespan} \leq X)$ is the maximum for a robust schedule.

We define the makespan with a normal distribution as $C_{max} \sim (\mu_{Cm}, \sigma_{Cm}^2)$ and so, the probability that makespan will not exceed the real maximum completion time limit X, is calculated as (21).

$$P(C_{max} \leq X) = 0.5 + \varphi(Z) \tag{21}$$

Where $Z = \frac{X - \mu_{Cm}}{\sigma_{Cm}}$, and $\varphi(Z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z} e^{-\left(\frac{t^2}{2}\right)} dt$.

Since the function $\varphi(Z)$ is difficult to get its exact solution, the mathematical relation provided by (Liu et al., 2011) is used as (22) to obtain its approximate solution as (22).

$$\phi(Z) \approx \varphi(Z) = \begin{cases} 0.1Z(4.4 - Z) & (0 \le Z \le 2.2) \\ 0.49 & (2.2 < Z < 2.6) \\ 0.5 & (Z \ge 2.6) \end{cases} \tag{22}$$

4-Solution approach

4-1-Genetic algorithm

In view of the fact that the problem at hand is strongly NP-hard and so, we need heuristic or metaheuristic algorithms to provide near solutions in a reasonable time. Genetic algorithm (GA) is one of the approximation optimization methods that utilize theories of evolution and natural selection to solve problems in a complex solution space and it is widely used for flow shop scheduling problems (Liu et al., 2011; Hasani & Hosseini, 2020). Therefore, an improved GA is tuned and used in this section to provide near-optimal solution for the problem at hand considering all aforementioned features.

Solution representation

The proposed algorithm uses permutation encoding, which is a common technique for solution representation in sequencing problems. Since the job-based representation provides a direct feasible solution in the presence of precedence constraints, this procedure is used for the considered problem. To this end, a population of main chromosomes is generated for sequencing the products and several sub-chromosomes are generated for parts assigning to the parallel machines. The best sub-chromosome for the main chromosome is selected based on the value obtained for the objective function. To simplify, it is assumed that if product h is precedes the product h', then, process operation of all parts of the product h' doesn't start before processing of all parts of the product h. For instance, suppose that four products are need to produce. The required part set of these products is as table 1. A sample of the encoding structure has been demonstrated in figure 2.

Table 1. The required part set of the instance

Product No.	1	2	3	4
Part No.	1,2	3,4,5	6,7	8,9

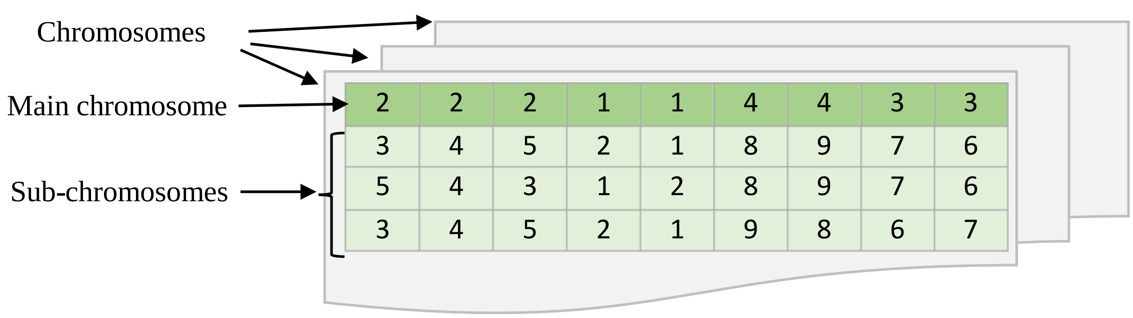

Fig 2. A sample of the proposed encoding structure

Initial population generation

Due to the notable importance of the initial solutions and their effect on the final result, it is better to some of the initial solutions are identified as suitable rules. Accordingly, we use two solutions provided by well-known heuristic schedules for flow shop. The first algorithm is NEH introduced by Nawaz, Enscore, and Ham in 1983 for scheduling of m-machine permutation flow shop (Nawaz et al., 1983)(Kalczynski &

Kamburowski, 2007). The second procedure is called JAF proposed by Hosseini et al. in 2021 for a two-stage assembly flow shop (Hosseini et al., 2021). Moreover, additional required initial solutions are generated randomly. These procedures are used for sequencing the products as the main chromosomes. After that, the set of parts sequencing is determined randomly for each product as the required sub-chromosomes.

Fitness function

In the problem at hand, due to the issue of uncertainty in assembling times, the fitness value of each chromosome i is evaluated as (23).

$$f(i) = P\left(C_{max}^{i} \leq X\right) \quad (23)$$

This function is calculated by considering (21) and (22). Obviously, the higher the fitness function value of an individual, the better the individual is to select.

Parent selection

Then, parents are selected by tournament selection method wherein, a tournament among all chromosomes is held to determine the selective pressure. This pressure forces the algorithm to select required parents with a large fitness value. The winner of the tournament is regarded as the best individual with the highest fitness value and is inserted into a mating pool of new offspring. The procedure is repeated until the mating pool of offspring is filled.

Crossover point

P1:

2			1		4		3	
2	2	2	1	1	4	4	3	3
3	4	5	2	1	8	9	7	6
5	4	3	1	2	8	9	7	6
3	4	5	2	1	9	8	6	7

Crossover point

P2:

1		2			3		4	
1	1	2	2	2	3	3	4	4
1	2	4	3	5	6	7	9	8
1	2	4	5	3	7	6	9	8
2	1	3	5	4	7	6	8	9

Crossover operator

C1:

2	2	2	1	1	3	3	4	4
3	4	5	2	1	6	7	9	8
5	4	3	1	2	7	6	9	8
3	4	5	2	1	7	6	8	9

C2:

1	1	2	2	2	4	4	3	3
1	2	4	3	5	8	9	7	6
1	2	4	5	3	8	9	7	6
2	1	3	5	4	9	8	6	7

Mutation operator

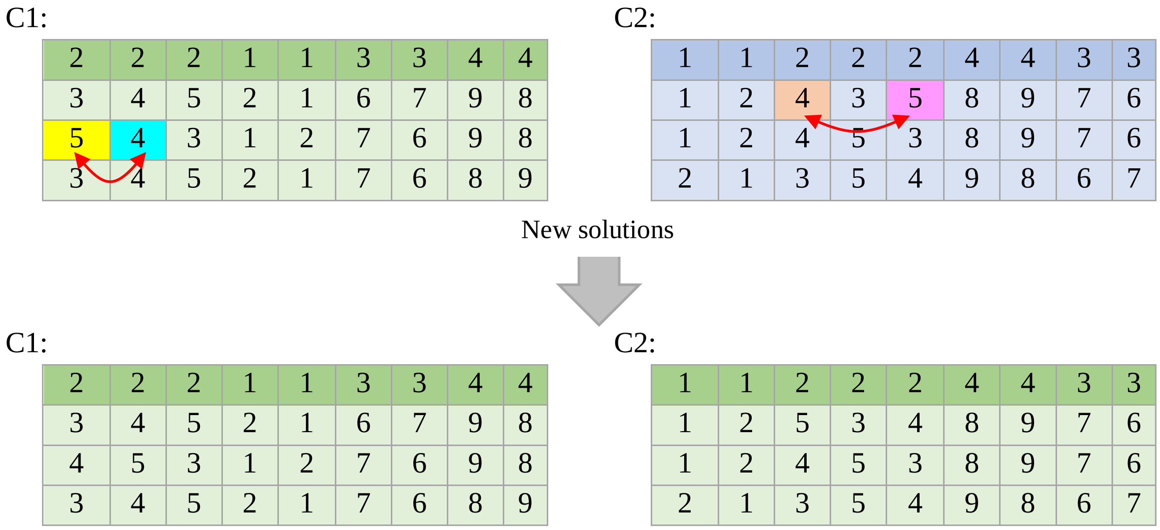

Fig 3. The proposed recombination operators

Crossover and mutation operators

The crossover operator is executed only on the main chromosome (the first string) that indicated products sequencing. Some different methods were tested for the crossover, and finally, the one-point crossover (1PX) was recognized well than the others. It should be noted that the cut point is determined randomly at the first of a product position on one of two chromosomes. Then the sub-chromosomes (the rest strings) that demonstrate the part sequencing are mutated according to the sequence of products and their assignment to the machines in the first stage. Based on test results, the swap operator is applied randomly on genes as the proper mutation. Figure 3 presents a sample of the one-point crossover and swap operator on parts sequencing used for the considered algorithm.

Replacement

For the problem at hand, the new generation P_{t+1} is selected from both parents and offspring based on the fitness values. In this way, the new generation is formed by replacing the second 50% of the current population with the first 50% of the new population (offspring). Based on this procedure, there is no guarantee that the inserted best solutions are always better than the worst solutions in the current population. This approach makes the algorithm towards searching the various regions and avoids forming a premature convergence during the progress of the algorithm.

Termination condition

Several termination conditions, such as a maximum number of generations, elapsed time, no change in fitness, and stall time limits were tested and finally applying two criteria of the maximum number of iterations and the number of successive iterations with no change in fitness simultaneously provided the best result. Figure 4 demonstrates the pseudo-code of the proposed genetic algorithm for the considered problem.

4-2-Simulated annealing

The simulated annealing (SA) algorithm as another well-known metaheuristic is tuned and used for result comparison. The same encoding structure explained for the proposed GA is used for the proposed SA. Moreover, three neighborhood structures including swapping, reversion, and insertion are applied randomly in each iteration to enhance the performance of the algorithm. Figure 5 represents a schematic view of the proposed SA.

Start: (Initialization)

Step 1: Create initial population

Step 2: Evaluate fitness of individuals

Step 3: Select parents (Tournament selection)

Step 4: Crossover (1PX)

Step 5: Mutation (swapping)

Step 6: Replacement (Both parent replacement)

Step 7: Create new generation with new generation scheme

If termination condition occurs (maximum number of iterations or the number of successive iterations with no change in fitness) proceed to the next step

Else, go to step 2

Step 8: Report the final result and stop

Fig 4. Main steps of the proposed GA

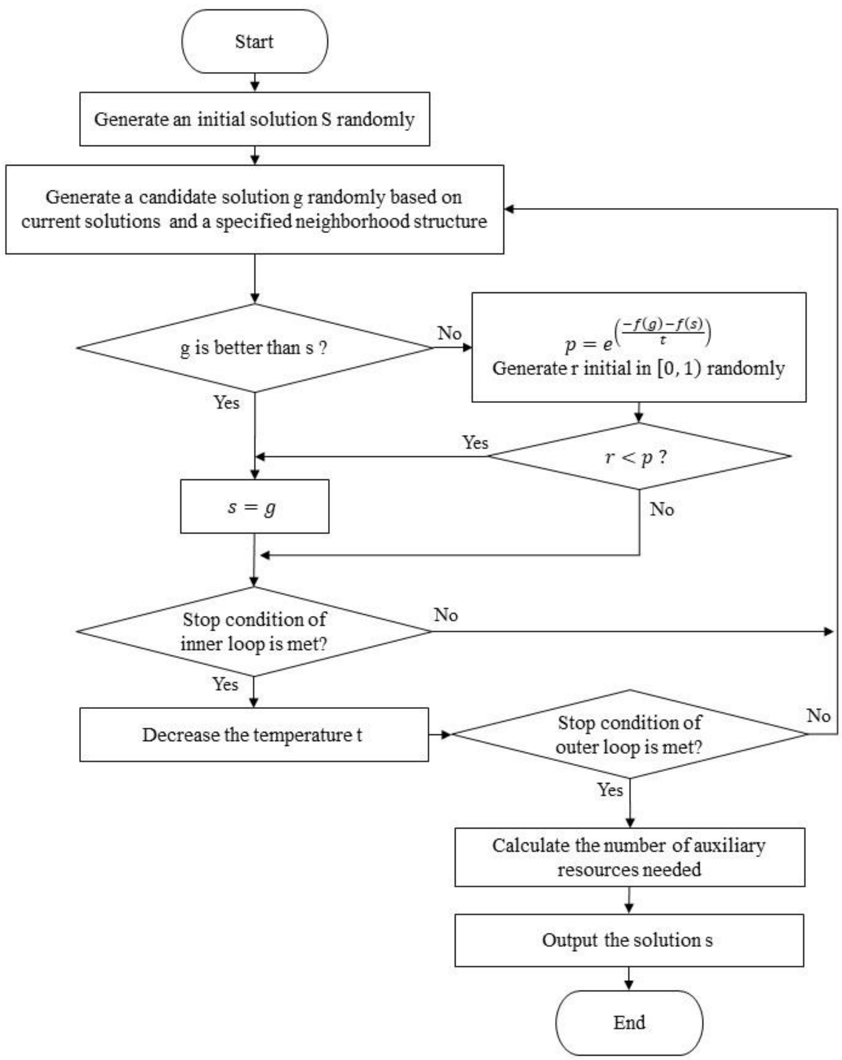

Fig 5. Flow chart of the used SA

4-3-Parameter calibration

Parameters tuning has a significant impact on the performance of approximation methods. Due to the number of required parameters, we should perform the parameter tuning using a proper design of experiments. The Taguchi method is a well-known experimental methodology that indicates the best values of required parameters with the minimum possible number of experiments. This method is one of the famous statistical analysis methods that can improve the robustness of an algorithm by obtaining a better parameter configuration (Yepes-Borrero et al., 2020). Therefore, this method is used in this section to determine the best combination of required parameters for the two abovementioned algorithms. Three levels are tested for each parameter and evaluation is performed by following the signal-to-noise (S/N) ratio concerning the independent variables. For simplicity, the procedure is handled under deterministic condition of the problem at hand. Since the objective in this study is to minimize the makespan, the smaller-the-better principle is considered as (24).

$$\frac{s}{n} = -10\log\frac{1}{n}\left(\sum x^2\right) \tag{24}$$

Where x is the experimental data of the dependent variable (i.e., makespan); n is the number of experimental observations. The main factors considered in this paper, parameter tuning data, and the results of the Taguchi method for each level of factors are shown in table 2.

Table 2. Tested and final values of parameters provided by Taguchi method

Algorithm	Parameters	Tested values			Final values
GA	Max iterations	250	400	550	400
	Population size (N)	50	70	100	100
	Crossover rates (Pc)	0.6	0.75	0.9	0.9
	Mutation rates (Pm)	0.05	0.15	0.2	0.15
SA	Initial temperature (T_0)	100	150	200	100
	Final temperature (T_f)	1×10^{-14}	1×10^{-9}	2×10^{-5}	1×10^{-14}
	Maximum inner iterations	5	10	15	10
	Temperature damping rate (α)	0.85	0.9	0.95	0.85

5-Computational experimentation

This section provides test instances and evaluation results of the performance of the proposed GA. In order to evaluate the robustness of proposed schedules, the Probability (makespan≤X) of schedules provided by the proposed algorithm is compared with another method and with the optimal solutions obtained from the mathematical model on the small scales.

To use the proposed algorithm for solving the problem at hand, first, some test instances are customized with inspiration from the standard data in the existing literature. To this end, data of test examples given by (Fattahi et al., 2013) for a two-stage hybrid flow shop scheduling problem with assembly operation are modified for the considered problem in this study and analyzed. Since, the available data in (Fattahi et al., 2013) only provides deterministic processing time of parts and assembling time of products, the deterministic assembling time of each product h is considered as the mean assembling times of that product (μ_{Ah}). The variance of assembling time corresponding to each product at the second stage is randomly taken from an assumed interval of $[1\ ,\ 0.1\mu_{Ah}]$. Table 3 demonstrates all characteristics of test examples.

Table 3. Characteristics of test instances

Instance name	H	n_H	m	P_j	μ_{Ah}	σ^2_{Ah}
AF1	10	5	2	50	200	[1 , 20]
AF2	10	5	3	50	200	[1 , 20]
AF3	10	5	4	50	200	[1 , 20]
AF4	10	7	2	50	200	[1 , 20]
AF5	10	7	3	50	200	[1 , 20]
AF6	10	7	4	50	200	[1 , 20]
AF7	10	10	2	50	200	[1 , 20]
AF8	10	10	3	50	200	[1 , 20]
AF9	10	10	4	50	200	[1 , 20]
AF10	50	5	2	50	200	[1 , 20]
AF11	50	5	3	50	200	[1 , 20]
AF12	50	5	4	50	200	[1 , 20]
AF13	50	7	2	50	200	[1 , 20]
AF14	50	7	3	50	200	[1 , 20]
AF15	50	7	4	50	200	[1 , 20]
AF16	50	10	2	50	200	[1 , 20]
AF17	50	10	3	50	200	[1 , 20]
AF18	50	10	4	50	200	[1 , 20]
AF19	100	5	2	50	200	[1 , 20]
AF20	100	5	3	50	200	[1 , 20]
AF21	100	5	4	50	200	[1 , 20]
AF22	100	7	2	50	200	[1 , 20]
AF23	100	7	3	50	200	[1 , 20]
AF24	100	7	4	50	200	[1 , 20]
AF25	100	10	2	50	200	[1 , 20]
AF26	100	10	3	50	200	[1 , 20]
AF27	100	10	4	50	200	[1 , 20]
AF28	150	5	2	50	200	[1 , 20]
AF29	150	5	3	50	200	[1 , 20]
AF30	150	5	4	50	200	[1 , 20]
AF33	150	7	2	50	200	[1 , 20]
AF34	150	7	3	50	200	[1 , 20]
AF35	150	7	4	50	200	[1 , 20]
AF36	150	10	2	50	200	[1 , 20]
AF37	150	10	3	50	200	[1 , 20]
AF38	150	10	4	50	200	[1 , 20]

To compare result of the proposed algorithms, we use Johnson's rule that has proven its superiority for two-stage flow shop scheduling problems. This idea was introduced in 1954 as an exact method for job sequencing in two-stage flow shop scheduling problems with minimizing makespan as the objective function.

Procedure of the Johnson's idea can be summarized as below:

Suppose that p_{i1} and p_{i2} are the processing time of job i at the first and second stage respectively. Similarly, p_{j1} and p_{j2} are considered as processing time of job j. In the optimal schedule, job i precedes job j if the following condition is met:

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

Several studies have extended this rule to the two-stage assembly flow shop sequencing problem; for instance, we can cite, (Fattahi et al., 2013), (Hosseini, 2016), and (Hosseini et al., 2021). So, we use the extended Johnson's rule to solve the problem at hand and to calculate the Probability (makespan≤X) of the proposed algorithm based on solutions provided by this technique. In order to compare Probability (makespan≤X), the expected completion time is needed. In this way, Probability (makespan≤X) for Johnson's rule schedules is assumed and the corresponding Z value is then calculated from equations (21) and (22).

The expected completion time of products is calculated based on the Z value. After that, the expected completion time is used to calculate the Probability (Makespan≤X) of the schedules provided by the proposed algorithm. Finally, this Probability of GA schedules is compared with the assumed Probability (Makespan≤X) of Johnson's idea.

First, we need to adopt the problem at hand for using Johnson's algorithm. In this way, the parallel machines at the processing stage are considered as the first stage and the assembly stage is considered as the second stage. The mean assembling time of each product h is considered as p_{h2}. Moreover, the total time for processing and completing the set of parts for each product (n_h) can be computed as p_{h1}. This time depends on some factors in addition to the processing times such as the number of the parts and the number of the parallel machines. So, the maximum amount of two below phrases is considered as the processing time of the parts for each product as below (Hosseini et al., 2021).

$$\max_{j \in n_h}(P_j) \ \forall h$$

$$\frac{\sum_{j \in n_h}(P_j)}{m} \ \forall h$$

The maximum amount of two above phrases is considered as p_{h1} and the assembling time of each production h indicates the p_{h2}. Now, the sequencing of the products is done using Johnson's idea as below steps:

- Suppose $U = \{h \in H | p_{h1} < p_{h2}\}$ and $V = \{h \in H | p_{h1} \geq p_{h2}\}$
- Sort the set of U in non-decreasing order of p_{h1} and set of V in non-increasing order of p_{h2}
- Determine the sequence of products according to the set of U and V after that

After sequencing the products, the parts of each product are sorted in non-increasing processing time to process by the parallel machines.

Table 4. Result of solving test instances

Problem	Johnson's Idea		GA			%Increase	%Decrease
	C_{max}	X	$\varphi(Z)$	Probability	C_{max}	Probability	Risk
AF1	2004	2031.1	0.5	1	1997	25	20
AF2	1829	1859.3	0.5	1	1829	25	20
AF3	1905	1945.4	0.492	0.992	1905	23	18.11
AF4	2183	2186.9	0.447	0.947	2174	11.75	8.5
AF5	2062	2106.5	0.5	1	2062	25	20
AF6	2036	2079.8	0.5	1	2036	25	20
AF7	2678	2694	0.455	0.955	2664	13.75	10.09
AF8	2651	2707.7	0.468	0.968	2651	17	12.78
AF9	2629	2685	0.465	0.965	2629	16.25	12.15
AF10	8615	8793.1	0.495	0.995	8568	23.75	18.81
AF11	8616	8800.4	0.49	0.990	8532	22.5	17.65
AF12	8636	8821.3	0.435	0.935	8571	8.75	6.19
AF13	9027	9220.5	0.459	0.959	8918	14.75	10.91
AF14	8847	9036.7	0.498	0.998	8768	24.5	19.52
AF15	8998	9180.7	0.419	0.919	8983	4.75	3.27
AF16	12062	12320.2	0.405	0.905	12016	1.25	0.84
AF17	12086	12345.1	0.468	0.968	12059	17	12.78
AF18	12042	12300	0.47	0.970	11934	17.5	13.21
AF19	8904	9094.5	0.459	0.959	8823	14.75	10.91
AF20	12077	12325.5	0.417	0.917	12009	4.25	2.92
AF21	12059	12317.5	0.301	0.801	11957	-24.75	-14.16
AF22	12093	12351.9	0.487	0.987	11926	21.75	16.96
AF23	16819	17179	0.439	0.939	16832	9.75	6.95
AF24	16862	17223.4	0.425	0.925	16763	6.25	4.35
AF25	16999	17363.2	0.498	0.998	16878	24.5	19.52
AF26	17995	18381	0.415	0.915	17889	3.75	2.56
AF27	17604	17981	0.428	0.928	17559	7	4.9
AF28	17495	17869.5	0.5	1	17261	25	20
AF29	23925	24437.3	0.496	0.996	23857	24	19.05
AF30	23355	23855.5	0.375	0.875	23343	-6.25	-4
AF31	23382	23883.1	0.481	0.981	23362	20.25	15.61
AF32	25570	26117.4	0.425	0.925	25554	6.25	4.35
AF33	25502	26048.8	0.5	1	25295	25	20
AF34	25465	26010.6	0.438	0.938	25399	9.5	6.76
AF35	26264	26796.5	0.392	0.892	26068	-2	-1.32
AF36	26188	26749.4	0.412	0.912	26027	3	2.04

All procedures are coded MATLAB (R2019b). Then experiments are executed on a PC with a 2.0GHz Intel Core 2 Duo processor and 2GB of RAM. Each problem is considered to have a probability value of 0.8 for schedules provided by Johnson's idea. The Z value relevant to this probability level is taken from the standard normal distribution table which is 1.209.

Every example, is first solved by the Johnson's idea to get average and variance makespan of schedule. Then, for each example the corresponding expected completion time is calculated as below and represented by X in table 4.

$$Z = \frac{X - \mu_{Cm}}{\sigma_{Cm}}$$

Furthermore, the mean and variance makespan for each instance is calculated for the schedules obtained from the proposed algorithm. In this way, the X values obtained from Johnson's idea of each instance are taken as such to represent the expected completion time for each instance of the schedule. The probability value of each instance is also determined for their corresponding schedules using (22). Finally, the percentage increase in probability values and percentage decrease in risk of each instance is computed as (25) and (26) respectively (Liu et al., 2011).

$$\%Increase\ Probability = \frac{\varphi(Z)_{alg.} - \varphi(Z)_{JI}}{\varphi(Z)_{JI}} \times 100 \quad (25)$$

$$\%Decrease\ Risk = \frac{\left(1 - \varphi(Z)_{JI}\right) - \left(1 - \varphi(Z)_{alg.}\right)}{\left(1 - \varphi(Z)_{alg.}\right)} \times 100 \quad (26)$$

Results shown in table 4 indicate that the proposed genetic algorithm gets better robust schedules for most examples except three instances (i.e., instances AF21, AF30, and AF35) than Johnson's procedure schedules. Results of six instances give 100% probability of ensuring that makespan will not cross the expected completion time limit. Minimum percentage increase in the probability of 3% is obtained with 2.04% decrease in risk from GA excluding three examples AF21, AF30, and AF35 that Johnson's method outperformed the proposed GA. Maximum percentage increase in the probability of 25% with a maximum decrease in risk of 20% is observed from the proposed GA.

To better evaluate the performance of the proposed genetic algorithm, the nine first examples are solved ten times by GA and SA and the result is compared with the optimal solutions provided by the mathematical model. In this way, the deviation percentage of the expected makespan from the optimal schedule is presented for two algorithms. This index is calculated as (27) in two cases as the average and maximum percentage increase over the optimal solution for the algorithms.

$$\%D = \frac{Z_{Alg.} - Z_{opt}}{Z_{opt}} \times 100 \quad (27)$$

Figures 6 and 7 demonstrate %D in the mean and maximum cases respectively. As it is depicted in these figures, GA has better performance in solving the problems in terms of both mean and maximum deviation from the optimal schedule. However, the result denotes that both of the two proposed algorithms provide the expected makespan of the robust schedule closely approximates the schedule with the optimal expected makespan. Genetic algorithm shows the mean deviation 3.2% to 14.9% from the optimal solutions in solving the problem on the small scales. However, simulated annealing has solved the problem with the average approximation errors of 5.6% to 28.7%.

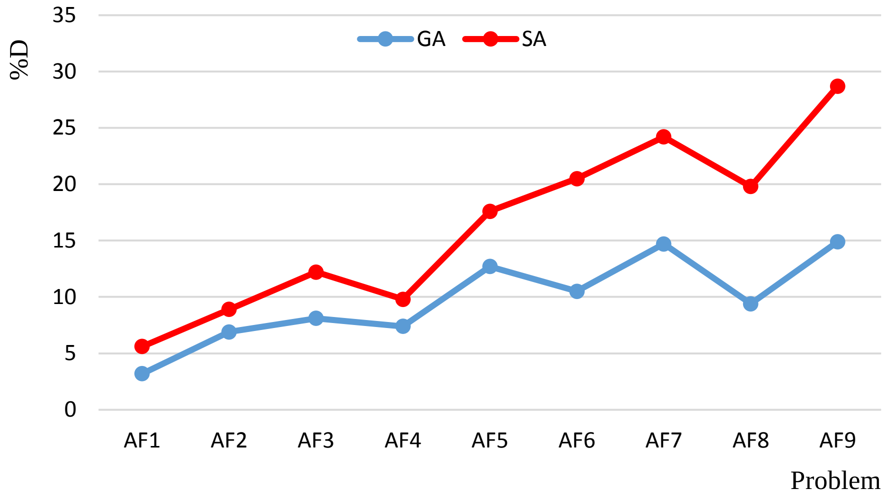

Fig 6. The average of %D

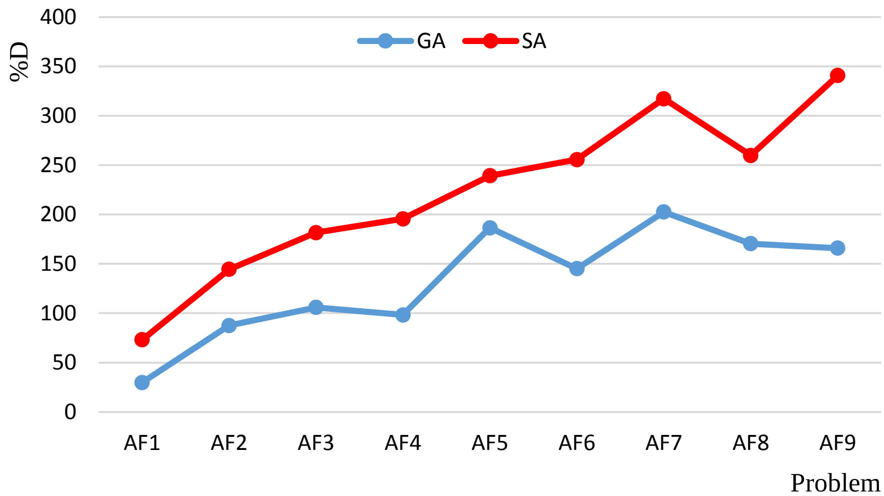

Fig 7. The maximum of %D

Figure 8 indicates the CPU time of two algorithms for solving the problem on different sizes. As expected, the CPU time of both two algorithms rises as the problem size increases. However, it can be concluded that simulated annealing consumes less CPU time than GA. The reason could be more power of GA in explore the solution space.

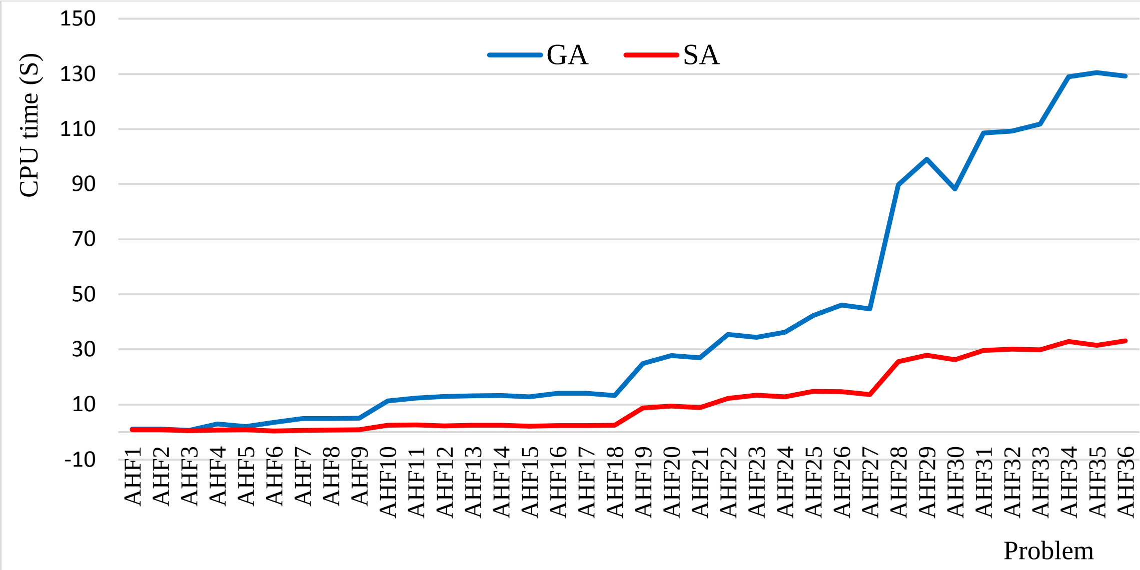

Fig 8. The CPU times of two algorithms

6-Conclusions

This study discussed the makespan minimisation-scheduling problem in a two-stage assembly flow shop under uncertainty in the assembling times. The first stage has m identical parallel machines to process components and the second stage is a workstation to assemble the components into the final products. Due to the conditions of the assembly machine and tools, different worker skill levels, and operational factors, it was assumed that there is uncertainty in assembling time to close the problem to the real-world condition. To this end, the assembling time distribution of products was considered as normally distributed and the main objective was to propose a robust schedule to maximize the probability of ensuring that makespan would not exceed the expected completion time.

The two-stage assembly flow shop scheduling problem has received more attention in recent years due to its applications in manufacturing enterprises. The majority of the existing studies in the literature consider the processing and assembling times as deterministic parameters. However, there are usually different factors that managers have to schedule the jobs under uncertainty in the processing and assembling times. Therefore, it is vital for managers to consider all practical features of the problem to find out practical solutions. To the best of our knowledge, this paper is the first attempt to deal with uncertainty in assembling times as a practical feature of the aforementioned problem.

The problem was described carefully and formulated using an MIP model and an improved genetic algorithm was proposed and adopted to get the robust schedule. In this way, a new solution representation in accordance with the problem features was proposed. Moreover, the value of Probability ($C_{max} \leq X$) for each chromosome was defined as a new fitness function of chromosomes. A proper procedure was developed based on Johnson's idea as a valid reference to evaluate the performance of the proposed algorithm. Furthermore, the schedules obtained from the proposed algorithm were compared with modified simulated annealing in different condition.

Some experiment analyses were performed on 36 test examples taken from the existing literature. The computational result indicated that robust schedules obtained by the proposed algorithm hedge effectively against uncertain assembling times while maintaining excellent expected makespan performance. The result showed that the proposed genetic algorithm gets better robust schedules than Johnson's rule for 33 test examples of 36 instances. Comparison the performance of two algorithms GA and SA demonstrated that the proposed improved GA outperforms SA in terms of optimality that was evaluated as deviation

percentage ($\%D$) of the expected makespan from the optimal schedule for the small-sized scales. Conversely, GA is more computationally expensive due to its exploration procedure.
The result of this study can be valuable for managers of two-stage or distributed manufacturing industries under uncertainty in the assembling times. Solution approaches proposed in this research help these managers provide robust schedule in the face of time changes during product assembly. It should be noted that providing robust schedule that hedge effectively against uncertain assembling times, help managers to reduce many operational costs such as deviation from due date and customer unsatisfying.

Future research can extend the proposed methods for other scheduling problems such as three-stage assembly flow shop or flow shop with assembly stage. Other possible extensions of this work include considering setup times or uncertainty in processing times in addition to assembling times. In real-world condition, the due date of products might be different. So, future research can be extended to consider robust scheduling when all products have different due dates.

References

Allahverdi, A., & Al-Anzi, F. S. (2006). A PSO and a Tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers & Operations Research, 33*(4), 1056–1080. https://doi.org/10.1016/j.cor.2004.09.002

Allahverdi, A., & Al-Anzi, F. S. (2008). The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *International Journal of Advanced Manufacturing Technology*, *37*, 166–177. https://doi.org/10.1007/S00170-007-0950-Y

Allahverdi, A., & Aydilek, H. (2010). Heuristics for the two-machine flowshop scheduling problem to minimise makespan with bounded processing times. *International Journal of Production Research*, *48*(21), 6367–6385. https://doi.org/10.1080/00207540903321657

Allahverdi, A., & Aydilek, H. (2015). The two stage assembly flowshop scheduling problem to minimize total tardiness. *Journal of Intelligent Manufacturing*, *26*, 225–237. https://doi.org/10.1007/S10845-013-0775-5

Balasubramanian, J., & Grossmann, I. E. (2002). A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Computers & Chemical Engineering*, *26*(1), 41–57. https://doi.org/10.1016/S0098-1354(01)00735-9

Billaut, J. C., Moukrim, A., & Sanlaville, E. (2008). *Flexibility and robustness in scheduling*. John Wiley.

Daneshamooz, F., Fattahi, P., & Hosseini, S. M. H. (2021). Mathematical modeling and two efficient branch and bound algorithms for job shop scheduling problem followed by an assembly stage. *Kybernetes*. https://doi.org/10.1108/K-08-2020-0521/FULL/HTML

Fattahi, P., Hosseini, S. M. H., & Jolai, F. (2013). A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *65*(5–8), 787–802. https://doi.org/10.1007/s00170-012-4217-x

Freeman, N. K., Melouk, S. H., & Mittenthal, J. . (2016). A scenario-based approach for operating theater scheduling under uncertainty. *Manufacturing and Service Operations Management*, *18*(2), 245–261. https://doi.org/10.1287/MSOM.2015.0557

Ghezail, F., Pierreval, H., & Hajri-Gabouj, S. (2010). Analysis of robustness in proactive scheduling: A graphical approach. *Computers & Industrial Engineering*, *58*(2), 193–198. https://doi.org/10.1016/j.cie.2009.03.004

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

426.

Goren, S., & Sabuncuoglu, I. (2008). Robustness and stability measures for scheduling: single-machine environment. *IIE Transactions*, *40*(1), 66–83. https://doi.org/10.1080/07408170701283198

Hasani, A., & Hosseini, S. M. H. (2020). A bi-objective flexible flow shop scheduling problem with machine-dependent processing stages: Trade-off between production costs and energy consumption. *Applied Mathematics and Computation*, *386*, 125533. https://doi.org/10.1016/j.amc.2020.125533

Herroelen, W., & Leus, R. (2004). Robust and reactive project scheduling: A review and classification of procedures. *International Journal of Production Research*, *42*(8), 1599–1620. https://doi.org/10.1080/00207540310001638055

Hosseini, S. M. H. (2016). Modeling the hybrid flow shop scheduling problem followed by an assembly stage considering aging effects and preventive maintenance activities. *International Journal of Supply and Operations Management*, *3*(1), 1215–1233. http://www.ijsom.com/article_2669.html

Hosseini, S. M. H., Sana, S. S., & Rostami, M. (2021). Assembly flow shop scheduling problem considering machine eligibility restrictions and auxiliary resource constraints. *International Journal of Systems Science: Operations and Logistics*, 1–17. https://doi.org/10.1080/23302674.2021.1942586

Kalczynski, P. J., & Kamburowski, J. (2007). On the NEH heuristic for minimizing the makespan in permutation flow shops. *Omega*, *35*(1), 53–60.

Kasperski, A., Kurpisz, A., & Zieliński, P. (2012). Approximating a two-machine flow shop scheduling under discrete scenario uncertainty. *European Journal of Operational Research*, *217*(1), 36–43. https://doi.org/10.1016/j.ejor.2011.08.029

Kazemi, H., Mazdeh, M. M., & Rostami, M. (2017). The two stage assembly flow-shop scheduling problem with batching and delivery. *Engineering Applications of Artificial Intelligence*, *63*, 98–107. https://doi.org/10.1016/j.engappai.2017.05.004

Komaki, G. M., Teymourian, E., Kayvanfar, V., & Booyavi, Z. (2017). Improved discrete cuckoo optimization algorithm for the three-stage assembly flowshop scheduling problem. *Computers & Industrial Engineering*, *105*, 158–173. https://doi.org/10.1016/j.cie.2017.01.006

Koulamas, C., & Kyparisis, G. j. (2001). The three-stage assembly flowshop scheduling problem. *Computers & Operations Research*, *28*(7), 689–704. https://doi.org/10.1016/S0305-0548(00)00004-6

Kouvelis, P., Daniels, R. L., & Vairaktarakis, G. (2000). Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions (Institute of Industrial Engineers)*, *32*(5), 421–432. https://doi.org/10.1023/A:1007640726040

Lan, S., Clarke, J. P., & Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, *40*(1), 15–28. https://doi.org/10.1287/TRSC.1050.0134

Lee, C. Y., Cheng, T. C. E., & Lin, B. M. T. (1993). Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. *Management Science*, *39*(5), 616–625. https://doi.org/10.1287/MNSC.39.5.616

Lei, D., Su, B., & Li, M. (2020). Cooperated teaching-learning-based optimisation for distributed two-stage assembly flow shop scheduling. *International Journal of Production Research*, 1–14. https://doi.org/10.1080/00207543.2020.1836422

Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, *32*(4–5), 715–727. https://doi.org/10.1016/j.compchemeng.2007.03.001

Liao, W., & Fu, Y. (2020). Min–max regret criterion-based robust model for the permutation flow-shop scheduling problem. *Engineering Optimization*, *52*(4), 687–700. https://doi.org/10.1080/0305215X.2019.1607848

Liu, Q., Ullah, S., & Zhang, C. (2011). An improved genetic algorithm for robust permutation flowshop scheduling. *The International Journal of Advanced Manufacturing Technology*, *56*(1–4), 345–354. https://doi.org/10.1007/s00170-010-3149-6

Navaei, J., Ghomi, S. M. T. F., Jolai, F., Shiraqai, M. E., & Hidaji, H. (2013). Two-stage flow-shop scheduling problem with non-identical second stage assembly machines. *The International Journal of Advanced Manufacturing Technology Volume*, *69*, 2215–2226. https://doi.org/10.1007/s00170-013-5187-3

Nawaz, M., Enscore, E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95. https://doi.org/10.1016/0305-0483(83)90088-9

Sabuncuoglu, I., & Goren, S. (2009). Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing*, *22*(2), 138–157. https://doi.org/10.1080/09511920802209033

Sung, C. S., & Juhn, J. (2009). Makespan minimization for a 2-stage assembly scheduling problem subject to component available time constraint. *International Journal of Production Economics*, *119*(2), 392–401. https://doi.org/10.1016/j.ijpe.2009.03.012

Sung, C. S., & Kim, A. H. (2008). A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times. *International Journal of Production Economics*, *113*(2), 1038–1048. https://doi.org/10.1016/j.ijpe.2007.12.007

Tang, L., & Wang, X. (2008). A predictive reactive scheduling method for color-coating production in steel industry. *The International Journal of Advanced Manufacturing Technology*, *35*, 633–645. https://link.springer.com/content/pdf/10.1007/s00170-006-0740-y.pdf

Torabzadeh, E., & Zandieh, M. (2010). Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flowshop. *Advances in Engineering Software*, *41*(10–11), 1238–1243. https://doi.org/10.1016/j.advengsoft.2010.06.004

Verderame, P. M., Elia, J. A., Li, J., & Floudas, C. A. (2010). Planning and scheduling under uncertainty: A review across multiple sectors. *Industrial and Engineering Chemistry Research*, *49*(9), 3993–4017. https://doi.org/10.1021/IE902009K

Wang, X., Wang, B., Zhang, X., Xia, X., & Pan, Q. (2021). Two-objective robust job-shop scheduling with two problem-specific neighborhood structures. *Swarm and Evolutionary Computation*, *61*, 100805. https://doi.org/10.1016/j.swevo.2020.100805

Wu, C. C., Bai, D., Azzouz, A., Chung, I. H., Cheng, S. R., Jhwueng, D. C., Lin, W. C., & Said, L. Ben. (2020). 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. *Engineering Optimization*, *52*(6), 1009–1036. https://doi.org/10.1080/0305215X.2019.1632303

Yepes-Borrero, J. C., Villa, F., Perea, F., & Caballero-Villalobos, J. P. (2020). GRASP algorithm for the unrelated parallel machine scheduling problem with setup times and additional resources. *Expert Systems with Applications*, *141*, 112959. https://doi.org/10.1016/j.eswa.2019.112959

Zhang, Z., & Tang, Q. (2021a). Integrating flexible preventive maintenance activities into two-stage assembly flow shop scheduling with multiple assembly machines. *Computers & Industrial Engineering*, *159*, 107493. https://doi.org/10.1016/j.cie.2021.107493

Zhang, Z., & Tang, Q. (2021b). Integrating preventive maintenance to two-stage assembly flow shop scheduling: MILP model, constructive heuristics and meta-heuristics. *Flexible Services and Manufacturing*

Journal, 1–48. https://doi.org/10.1007/S10696-021-09403-0

Zheng, P., Zhang, P., Wang, J., Zhang, J., Yang, C., & Jin, Y. (2020). A data-driven robust optimization method for the assembly job-shop scheduling problem under uncertainty. *International Journal of Computer Integrated Manufacturing*, 1–16. https://doi.org/10.1080/0951192X.2020.1803506

Ziaei, Z., & Jabbarzadeh, A. (2021). A multi-objective robust optimization approach for green location-routing planning of multi-modal transportation systems under uncertainty. *Journal of Cleaner Production*, *291*(0959–6526), 125293. https://doi.org/10.1016/j.jclepro.2020.125293