Scheduling on Flexible Flow Shop with Cost-related Objective Function Considering Outsourcing Options

Mojtaba Enayati\(^a\), Ebrahim Asadi-Gangraj\(^a\), Mohammad Mahdi Paydar\(^a\)

\(^a\) Babol Noshirvani University of Technology; Babol; Iran.

Received 15 July 2019; Revised 12 October 2020; Accepted 25 October 2020

Abstract

This study considers outsourcing decisions in a flexible flow shop scheduling problem, in which each job can be processed by either an in-house production line or outsourced. The selected objective function aims to minimize the weighted sum of tardiness costs, in-house production costs, and outsourcing costs with respect to the jobs due date. The purpose of the problem is to select the jobs that must be processed in-house, schedule processing of the jobs in-house, and finally select and assign other jobs to the subcontractors. We develop a mixed-integer linear programming (MILP) model for the research problem. Regarding the complexity of the research problem, the MILP model cannot be used for large-scale problems. Therefore, four metaheuristic algorithms, including SA, GA, PSO, hybrid PSO-SA, are proposed to solve the problem. Furthermore, some random test problems with different sizes are generated to evaluate the effectiveness of the proposed MILP model and solution approaches. The obtained results demonstrate that the GA can obtain better solutions in comparison to the other algorithms.

Keywords: Flexible flow shop scheduling; Outsourcing; Cost-related objective functions; Metaheuristic algorithms

1. Introduction

Scheduling problem is a branch of the operations research that is widely studied during the last years. It is one of the main activities in some production and service systems (Hosseini, 2019). Scheduling aims to allocate finite resources to the jobs to optimize the objective functions (Behnamian, 2020). It refers to the problems in manufacturing systems, in which, the jobs must be scheduled to process on one or more machines concerning a wide range of the objective functions. Regarding the job features, machines layout in the production lines, processing constraints, and objective functions, there is a large variety of scheduling problems in the machine scheduling domain.

The flow shop scheduling problem is a common manufacturing system, so that it is considered by some researchers since Johnson’s seminal paper (Johnson, 1954). In the flow shop problem, several stages are located in series, where there is only one machine in each stage (Nahavandi and Asadi-Gangraj, 2014). Nowadays, to increase the capacity, balance the capacities in a particular stage, and increasing demand for some products, it has been led to consider parallel machines in some stages in many companies. This developed environment is called as flexible flow shop (FFS), hybrid flow shop (HFS), flexible flow line (FFL), hybrid flow line (HFL), or flow shop with multiprocessor. In the FFS environment, machines are arranged into \( m \) stages in series, so that in stage \( i \), there are \( S_i \) unrelated parallel machines (Asadi-Gangraj, 2018).

The FFS problem combines two types of well-known scheduling problems: parallel machine scheduling problem and flow shop scheduling problem. The purpose of the FFS is to specify two decisions: sequence of the jobs through the shop-floor and allocation of the jobs to machines. Hence, the FFS scheduling problem tries to determine assignment of the jobs to the machines and scheduling them in each stage.

Nowadays, most companies use outsourcing option in their industries, where they give some jobs to subcontractors. Via proper outsourcing decision, different costs of the company, such as operating costs, inventory costs, and delivering costs, can be reduced and it also leads to the company to be more flexible. With respect to the globalization and information technology growth, outsourcing plays a key role in manufacturing systems and gives some benefits to the companies in different manners. Through outsourcing some no-serious tasks to the subcontractors, the company can concentrate more on its main tasks.

In this paper, an FFS scheduling problem is studied where each job must be processed by in-house production line or subcontractor production line. The objective of the problem is to minimize tardiness costs, in-house production costs, and outsourcing costs. The main purpose of the problem is to make a subset of jobs that are associated with the in-house production line then scheduling these jobs, and make a subset of jobs that are associated with subcontractors and select a subcontractor for each outsourced job.

*Corresponding author Email address: e.asadi@nit.ac.ir
The remaining of the paper is structured as follows. A brief review of the related studies is provided in section 2. In Section 3, the proposed MILP model is described. Section 4 presents the proposed metaheuristic algorithms. Section 5 is devoted to provide computational experiments and finally, conclusions and future research are presented in section 6.

2. Literature Review

There are many studies in simple environments such as single machine and parallel machine environments about the scheduling problems with outsourcing options. Ruiz-Torres et al. (2006) presented a parallel machine scheduling problem with outsourcing option. The objective function minimizes outsourcing time and cost, and the number of tardy jobs. Lee and Sung (2008) dealt with a single machine scheduling problem by considering the outsourcing, where jobs must be either processed in-house or outsourced. The selected objective function is to minimize the weighted sum of the completion times and sum of the outsourcing cost. They also provided two heuristic approaches and a branch and bound (B&B) algorithm for this problem. Lee and Sung (2008) presented two scheduling problems in a single machine environment that outsourcing is allowed. The objective is to minimize maximum lateness and total tardiness considering outsourcing budget. Because of NP-hardness, they proposed two heuristic approaches and a B&B algorithm. Qi (2008) studied a scheduling problem with outsourcing in a single machine environment. There is a subcontractor with single machine environment and outsourced orders must ship back to the in-house shop in batches. The objective aims to minimize the weighted sum of makespan and total outsourcing and transportation cost. He proposed a dynamic programming approach for this problem. Chen and Li (2008) considered a parallel machine scheduling problem with outsourcing options with total production and outsourcing costs minimization. They proposed a heuristic algorithm for this problem. Mishra et al. (2008) developed an MILP model for the integrated planning and scheduling with respect to outsourcing supply chain. They presented a method to make strategic decisions. Chan et al. (2009) developed an enhanced swift converging simulated annealing algorithm for the scheduling problem with outsourcing option. The proposed algorithm is evaluated by comparing with other metaheuristic methods, GA, SA, TS and TS-SA algorithms. The results showed that this algorithm has a better performance to choose the best subcontractor for the jobs. Haoues et al. (2013) considered a scheduling problem to deal simultaneously with the in-house scheduling and outsourcing cost. They developed a GA-based algorithm to handle the research problem. Mokhtari and Abadi (2013) developed a mathematical model for planning the in-house and outsourced jobs, simultaneously. The selected objective function minimizes sum of the total weighted completion time and total outsourcing cost. Zhong and Huo (2013) presented a single machine scheduling problem with outsourcing option, in which each job can be processed within the in-house production line or to be outsourced to the subcontractors. They considered two objective functions for this problem. Neto et al. (2015) studied a parallel machine scheduling problem by considering the outsourcing option. They proposed an ant colony optimization algorithm to solve this problem. The selected objective function aims to minimize sum of the outsourcing and tardiness costs. Choi et al. (2016) presented a mini-max regret of a single machine scheduling problem to determine which jobs should be outsourced to the subcontractors with total in-house and outsourcing production costs minimization. They proposed heuristic algorithms to solve this problem. Also, some studies are presented in a more complicated environment such as flow shop or job shop environments. Lee et al. (2002) developed a mathematical model for advanced planning and scheduling problem with respect to the outsourcing options. Also, they developed a GA-based heuristic approach to solve this problem. Chung et al. (2005) presented a job shop scheduling problem with outsourcing to satisfy the due dates. They developed a heuristic algorithm for this problem. Qi (2009) dealt with makespan minimization in a two-stage flow shop with outsourcing options, so that only some of the operations can be outsourced. Lee and Choi (2011) dealt with a two-stage flow shop scheduling problem, in which each job can be processed by using in-house production line or outsourced to the subcontractor. The selected objective minimizes the weighted sum of the makespan and the total outsourcing costs. Neto and Filho (2011) proposed two independent ant colony optimization algorithms for the scheduling problem in permutation flow shop environment with outsourcing with respect to minimizing the makespan and outsourcing costs. Choi and Chung (2011) studied a two-machine flow shop scheduling problem with outsourcing options in order to minimize makespan and total outsourcing costs. They developed a polynomial-time algorithm to solve the problem. Qi (2011) presented a two-stage flow shop scheduling problem with respect to options of outsourcing some operations to the subcontractors to minimize the makespan. They considered different modes of outsourcing and proposed optimization algorithms for each mode. Moghaddam et al. (2012) presented a mathematical model for a two-machine flow shop scheduling problem with outsourcing options. The objective function is to minimize the total completion time for the in-house jobs and outsourcing cost. They developed a genetic-based approach to solve large-size problems. Mokhtari et al. (2012) dealt with a multi-stage flow shop scheduling problem, where the outsourcing option is allowed. The objective of the problem is to minimize the sum of weighted flow time, transportation cost, and processing time of outsourced jobs. They also proposed an MILP model and a team process.
algorithm. Chung et al. (2013) studied a two-machine flow shop scheduling problem with the outsourcing option to minimize the sum of the makespan and the total outsourcing cost. They developed an approximation algorithm for this problem. Guo et al. (2014) studied a bi-objective job shop scheduling problem considering the outsourcing, with respect to total outsourcing and tardiness cost. They used a lexicographic approach to consider these objectives, simultaneously, and proposed a two-phase neighborhood search to solve the problem. Lei et al. (2016) presented a job shop scheduling problem with outsourcing options. The objective function minimizes total tardiness regarding the limited outsourcing budget. They proposed a novel shuffled frog leaping algorithm for this problem. Ahmadizar and Amiri (2018) dealt with outsourcing in a two-machine flow shop scheduling problem to minimize the sum of the makespan and outsourcing and transportation costs. They presented two mathematical models and an ACO algorithm for this problem. Tirkolaee et al. (2020) studied outsourcing option and Just-in-Time delivery in the slow shop scheduling environment. They proposed a bi-objective MILP model and hybrid version of interactive fuzzy solution technique and a Self-Adaptive Artificial Fish Swarm Algorithm to minimize the total cost of the production system and total energy consumption. Wand and Cui (2020) considered robust identical parallel machine scheduling problem with outsourcing option and uncertain processing time. The selected objective function aims to minimize total completion time of in-house jobs and the cost of outsourcing jobs. They developed two approximation algorithms for the problem with discrete and interval scenarios.

In order to convenience the reader, some of the main researches in the context of scheduling problem with outsourcing are summarized in Table 1.

<table>
<thead>
<tr>
<th>Year</th>
<th>Author(s)</th>
<th>Environment</th>
<th>Objective Function</th>
<th>Solving Method</th>
<th>Number of subcontractors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>Lee et al.</td>
<td>Flow shop</td>
<td>Makespan</td>
<td>GA-based approach</td>
<td>One</td>
</tr>
<tr>
<td>2005</td>
<td>Chung et al.</td>
<td>Job shop</td>
<td>Outsourcing cost</td>
<td>Heuristic algorithms</td>
<td>One</td>
</tr>
<tr>
<td>2006</td>
<td>Ruiz-Torres et al.</td>
<td>Parallel machine</td>
<td>Total external machine utilization and the total number of late jobs</td>
<td>Heuristic algorithms</td>
<td>One</td>
</tr>
<tr>
<td>2008</td>
<td>Chen and Li</td>
<td>Parallel machine</td>
<td>Production and outsourcing costs</td>
<td>Heuristic algorithms</td>
<td>Multiple</td>
</tr>
<tr>
<td>2009</td>
<td>Qi</td>
<td>Flow shop</td>
<td>Outsourcing cost and makespan</td>
<td>Heuristic algorithms</td>
<td>one</td>
</tr>
<tr>
<td>2009</td>
<td>Chan et al.</td>
<td>Parallel machine</td>
<td>Makespan</td>
<td>GA, SA, and Fuzzy Logic Controller (FLC)</td>
<td>one</td>
</tr>
<tr>
<td>2011</td>
<td>Lee and Choi</td>
<td>Flow shop</td>
<td>Makespan and outsourcing cost</td>
<td>Heuristic algorithms</td>
<td>one</td>
</tr>
<tr>
<td>2012</td>
<td>Moghaddam et al.</td>
<td>Flow shop</td>
<td>Makespan and outsourcing cost</td>
<td>GA-based algorithm</td>
<td>One</td>
</tr>
<tr>
<td>2012</td>
<td>Moghtari et al.,</td>
<td>Flow shop</td>
<td>Flow time and outsourcing cost and transportation cost</td>
<td>Team process</td>
<td>Multiple</td>
</tr>
<tr>
<td>2013</td>
<td>Moghtari and Abadi</td>
<td>Parallel machine</td>
<td>Completion time and outsourcing cost</td>
<td>Heuristic algorithms</td>
<td>Multiple</td>
</tr>
<tr>
<td>2013</td>
<td>Chung and Choi</td>
<td>Flow shop</td>
<td>Makespan and outsourcing cost</td>
<td>Heuristic algorithms</td>
<td>One</td>
</tr>
<tr>
<td>2014</td>
<td>Guo and Lei</td>
<td>Job shop</td>
<td>Total tardiness and outsourcing cost</td>
<td>Heuristic algorithms</td>
<td>One</td>
</tr>
<tr>
<td>2015</td>
<td>Neto et al.</td>
<td>Parallel machine</td>
<td>Sum of outsourcing and delay costs</td>
<td>Ant colony optimization</td>
<td>One</td>
</tr>
<tr>
<td>2016</td>
<td>Lei and Guo</td>
<td>Job shop</td>
<td>Tardiness and outsourcing cost</td>
<td>Shuffled frog-leaping algorithm</td>
<td>One</td>
</tr>
<tr>
<td>2016</td>
<td>Choi and Chung</td>
<td>Single machine</td>
<td>Total production cost</td>
<td>Heuristic algorithms</td>
<td>One</td>
</tr>
<tr>
<td>2018</td>
<td>Ahmadizar and Amiri</td>
<td>Flow shop</td>
<td>Makespan, transporting and outsourcing cost</td>
<td>ACO-based algorithm</td>
<td>Two</td>
</tr>
<tr>
<td>2020</td>
<td>Tirkolaee et al.</td>
<td>Flow shop</td>
<td>Total cost of the production and total energy consumption,</td>
<td>Hybrid version of interactive fuzzy solution technique and a Self-Adaptive Artificial Fish Swarm Algorithm</td>
<td>One</td>
</tr>
<tr>
<td>2020</td>
<td>Wang and Cui</td>
<td>Identical parallel machine</td>
<td>Total completion time of in-house jobs and the cost of outsourcing jobs</td>
<td>Two approximation algorithms</td>
<td>One</td>
</tr>
<tr>
<td>2020</td>
<td>Present research</td>
<td>Flexible flow shop</td>
<td>Tardiness costs, in-house production costs, outsourcing costs</td>
<td>SA, GA, PSO, hybrid PSO-SA</td>
<td>Multiple</td>
</tr>
</tbody>
</table>
2.1. Research gap and contributions

The presented study investigates the flexible flow shop scheduling problem with unrelated parallel machines concerning outsourcing option. Three cost related objective functions, including tardiness costs, in-house production costs, and outsourcing costs, are considered in this research. In order to tackle this problem, a mixed-integer linear programming model is introduced. Besides, regarding the NP-hardness of this problem, four metaheuristic approaches, namely SA, GA, PSO, hybrid PSO-SA are proposed to solve this problem. With respect to the previous section and Table 1, the main contributions of this study are summarized in the following:

- The present research is the first try to consider the outsourcing option in the flexible flow shop environment with unrelated parallel machines.
- Multiple subcontractors are rarely dealt with in the scheduling problems with outsourcing options. As result, we consider this issue in the present research.
- Most of the papers in this context are used heuristic approaches for the scheduling problem with outsourcing option. Hence, four metaheuristic approaches are proposed to tackle the research problem, in which, a new heuristic approach is applied in the body of the metaheuristics to 1) divide the jobs into two subsets, including in-house jobs and outsourced jobs, 2) schedule the jobs in the in-house production line, 3) assign the outsourced jobs to the subcontractors, and finally 4) schedule the jobs on the subcontractor.

- A wide range of objective functions, including tardiness costs, in-house production costs, and outsourcing costs is considered in this paper.

3. Problem Description

There are \( n \) different jobs \((j = 1, \ldots, n)\), in which each job must be processed on in-house production line or outsourced. In-house machine layout is considered as flexible flow shop, in which, there some machines, which are arranged into \( s \) stages in series and there are \( k \) \((k = 1, \ldots, m_t)\) unrelated machines in parallel in each stage. Each machine can process only one job at a time and each job can be processed by only one machine at a time. Entire jobs have to be processed on only one machine at each stage and they are available at time zero. The travel time is neglected between the stages and setup time is considered in the processing times. Because of unrelated parallel machines at each stage, jobs processing time are different with respect to the different machines. Preemption is not allowed and there is unlimited storage space between the two successive stages.

Also, there are \( h \) \((h = 1, \ldots, H)\) different subcontractors, where each job that selected for outsourcing is fully outsourced to them. Each outsourced job is processed by one of the available subcontractors. The objective is to find an integrated schedule for the in-house and outsourced jobs to minimize the tardiness costs, in-house production costs, and outsourcing costs.

A schematic of the research problem is presented in Figure 1.

![Fig. 1. A schematic of the research problem](image-url)
3.1. Mathematical model

In this section, we will introduce parameters and decision variables to propose the mixed-integer linear programming (MILP) model for the research problem.

**Parameters**

The necessary parameters to present the proposed mathematical model are as follows:

- \( s \): Number of stages
- \( n \): Number of jobs
- \( H \): Number of subcontractors
- \( i \): Stage index
- \( j, l \): Job index
- \( h \): Subcontractor index, \( h \in (1, ..., H+1) \).
- \( H+1 \): Index of in-house production environment
- \( m_i \): Number of machines in stage \( i \)
- \( P_{ikj} \): Processing time of job \( j \) on machine \( k \) in stage \( i \)
- \( I_{Cj} \): In-house production cost of job \( j \)
- \( t_h \): Transportation time from in-house production line to subcontractor \( h \)
- \( PO_{hj} \): Processing time of job \( j \) on subcontractor \( h \)
- \( TC_j \): Tardiness cost of job \( j \)
- \( OC_{hj} \): Outsourcing cost of job \( j \) on subcontractor \( h \)
- \( M \): A big number

**Decision variables**

\( T_j \): Tardiness of job \( j \)
\( C_{ij} \): Completion time of job \( j \) on stage \( i \)
\( CO_{hj} \): Completion time of job \( j \) on subcontractor \( h \).
\( X_{ikj} \): if job \( j \) at stage \( i \) is assigned to machine \( k \), 1, otherwise, 0.
\( Y_{ijj} \): if job \( l \) is processed after job \( j \) at stage \( i \), 1, otherwise, 0.
\( Z_{hj} \): if job \( j \) assign to subcontractor \( h \), 1, 0, otherwise. If job \( j \) assign to in-house shop, \( z_{hH+1} = 1 \).
\( YO_{ijh} \): if job \( j \) is processed after job \( l \) on subcontractor \( h \), 1, otherwise, 0.

**MILP model**

The formulation of the MILP problem is as follows:

\[
\begin{align*}
\text{Min} \quad & \sum_{j} \sum_{h=H+1}^n \sum_{j} \sum_{j} \sum_{j} (1) \\
\sum_{h=H+1}^n (2) & \sum_{h=H+1}^n (3) \\
\sum_{h=H+1}^n (4) & \sum_{h=H+1}^n (5) \\
\sum_{h=H+1}^n (6) & \sum_{h=H+1}^n (7) \\
\sum_{h=H+1}^n (8) & \sum_{h=H+1}^n (9) \\
\sum_{h=H+1}^n (10) & \sum_{h=H+1}^n (11) \\
\sum_{h=H+1}^n (12) & \sum_{h=H+1}^n (13) \\
\sum_{h=H+1}^n (14) & \sum_{h=H+1}^n (15)
\end{align*}
\]
The objective function (1) aims to minimize the total costs, including total tardiness costs, in-house production costs, and outsourcing costs. Constraint set (2) determines the completion time of in-house jobs in the first stage and constraint set (3) calculates the completion time of in-house jobs in the other stages. Constraint sets (4) and (5) preclude the interference between the processing operations of any two jobs, which are processed on one machine in each stage. In a moment, at most one of these two constraint sets is activated. If jobs \( j \) and \( l \) are processed on machine \( k \) in stage \( i \) \((X_{ikj} = X_{ikl} = 1)\) and \( j \) is sequenced before \( l \) \((Y_{ij} = 1)\), constraint set 4 is activated. On the other side, if job \( l \) is sequenced before \( j \) \((Y_{ij} = 0)\), constraint set 5 will be activated. Finally, if jobs \( j \) and \( l \) are sequenced on different machines \((X_{ikj} + X_{ikl} \leq 1)\), both constraint sets will be redundant. Constraint set (6) ensures that in-house completion times of the outsourced jobs must be equal to 0. Constraint set (7) enforces that each job must be processed in an in-house production line or outsourced to one of the subcontractors. Constraints set (8) indicates that in-house jobs must be assigned to only one machine at each stage. Constraints set (9) calculates completion time of outsourced jobs regarding the transportation time from in-house production line to the subcontractor and processing time of the subcontractor. Constraint sets (10) and (11) consider the completion time of jobs \( j \) and \( l \) if they are outsourced to a subcontractor. Constraint set (12) indicates that completion times of in-house jobs on the subcontractors equal to 0. Constraint sets (13) and (14) determine tardiness of the in-house and outsourced jobs, respectively. Finally, constraint sets (15) and (16) show the range of decision variables.

### 4. Solutions Approaches

Since the flexible flow shop problem with unrelated machines is NP-hard (Aadi-Gangraj, 2018) in strong sense, the flexible flow shop problem with outsourcing option is also NP-hard. Therefore, the problems with medium to large size cannot be solved through the exact methods in a reasonable time; hence, in this study, we applied four metaheuristic approaches. For this purpose, four metaheuristic approaches, including simulated annealing (SA), genetic algorithm (GA), particle swarm optimization algorithm (PSO) and a hybrid version of SA and PSO (PSO-SA) algorithm, will be introduced.

#### 4.1. Simulated annealing algorithm

SA algorithm is firstly introduced by Kirkpatrick, Gelatt, and Vecchi (1983) and Cerny (1985) and it is one of the frequently used algorithms in the optimization problems. This approach is inspired by a procedure that includes heating and controlled cooling of a material to get the bigger size of its crystals and decrease their fault (Nayeri, et al. 2019). In order to solve an optimization problem, the SA algorithm first starts with an initial solution, and then moves to the neighboring solutions in a repeating loop. If the new solution is better than the current solution, the algorithm selects it as the best solution. Otherwise, the algorithm accepts the new solution with the probability \( \exp(-\Delta E/T) \) (Boltzmann operator) as the best solution. In this formula, \( \Delta E \) shows the difference between the objective function of new solution and the current solution, and \( T \) is the current temperature. Several iterations are performed in each temperature and the temperature is reduced, slowly. This process is performed until the stopping criteria have met. Regarding the literature, the SA algorithm has been successfully applied to a wide range of the scheduling problems. Following, different components of the SA algorithm is introduced.

#### 4.1.1. Solution structure

The first step in solving a problem with metaheuristic methods is to create an appropriate solution structure. The initial solution structure which is used in this algorithm is a vector in size of \( 1^n \) (number of jobs), so that each number represents a job and the vector presents a sequence of jobs.

#### 4.1.2. Create a neighborhood solution

In the SA algorithm, different methods are applied to create neighborhood solutions, such as swap, reversion, and insertion. In this research, by using a roulette wheel, these methods have been applied, randomly.

**Swap:** In the swap method, two genes of the sequence vector are selected randomly and the corresponding jobs are swapped. Figure 2 depicts an example of the swap with nine jobs.

Before Swap | 1 | 4 | 6 | 3 | 2 | 5 | 8 | 9 | 7  
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---  
After Swap | 1 | 4 | 8 | 3 | 2 | 5 | 6 | 9 | 7  

Fig. 2. An example of swap

**Reversion:** In the reversion method, by selecting two genes of the chromosome, the sequence of genes between them is reversed. An example of the revision method is showed in Figure 3.

Before Reversion | 1 | 4 | 6 | 3 | 2 | 5 | 8 | 9 | 7  
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---  
After Reversion | 1 | 4 | 8 | 5 | 2 | 3 | 6 | 9 | 7  

Fig. 3. An example of reversion
Insertion: Besides the aforementioned method, in the insertion method, two genes are firstly selected, and the second gene is removed and added after the first gene. This method can be seen in Figure 4.

Before Insertion: 1 6 4 3 2 5 7 9 8

After Insertion: 1 6 4 7 3 2 5 9 8

Fig. 4. An example of insertion

4.2. Genetic Algorithm (GA)

Genetic algorithm (Goldberg and Holland, 1988) is an iterative algorithm such that natural evolution is applied to model the search method. GA is one of the most widely used metaheuristic approaches for different optimization problems.

At the beginning of the GA, a random population is created and evaluated. Then, a percentage of the population (chromosome) is selected as the parent and children's population are formed by the combination of them. Similarly, a percentage of the population is selected for the mutation operations and the mutated population is generated. In the next step, the main population, children and mutated population are merged and the new main population is generated. If the stopping condition is fulfilled, the algorithm ends. Otherwise, the process will be repeated.

Table 1
Parameters and their values

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Description</th>
<th>Legend</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MaxIt</td>
<td>Maximum iteration</td>
<td>A</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>SA</td>
<td>MaxSubIt</td>
<td>Maximum of sub-iteration</td>
<td>B</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>Initial temperature</td>
<td>C</td>
<td>500</td>
<td>700</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>Alpha</td>
<td>Temperature reduction rate</td>
<td>D</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>GA</td>
<td>MaxIt</td>
<td>Maximum iteration</td>
<td>A</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Popsize</td>
<td>Population size</td>
<td>B</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Cross Rate</td>
<td>Crossover rate</td>
<td>C</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>Mut Rate</td>
<td>Mutation rate</td>
<td>D</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>PSO</td>
<td>MaxIt</td>
<td>Maximum iteration</td>
<td>A</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>PopSize</td>
<td>Population size</td>
<td>B</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>c₁</td>
<td>perceptual factor</td>
<td>C</td>
<td>0.75</td>
<td>0.85</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>c₂</td>
<td>Social factors</td>
<td>D</td>
<td>0.95</td>
<td>0.85</td>
<td>0.75</td>
</tr>
<tr>
<td>PSO-SA</td>
<td>MaxIt</td>
<td>Maximum iteration</td>
<td>A</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>PopSize</td>
<td>Population size</td>
<td>B</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>c₁</td>
<td>perceptual factor</td>
<td>C</td>
<td>0.75</td>
<td>0.85</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>c₂</td>
<td>Social factors</td>
<td>S</td>
<td>0.95</td>
<td>0.85</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>Initial temperature</td>
<td>E</td>
<td>500</td>
<td>700</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>Alpha</td>
<td>Temperature reduction rate</td>
<td>F</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Signal-to-noise (S/N) ratio is used to select the best levels of the metaheuristic parameters. The signal-to-noise graphs for the experiments are shown in Figures 8-11.

4.2.1 Solution structure

One of the main steps of the GA is to represent the solutions as a chromosome (Rezaei and Zarook, 2018). The chromosome is used to represent the solution that is similar to the one which is introduced for the SA algorithm. In order to correctly apply the operators of the GA on the chromosome, random numbers are generated in the interval of [0,1] and assign to each cell. Then, the numbers in the sequence vector are ranked and entered in the same vector. An example of the sequence vector for nine jobs is showed in Figure 5.

<table>
<thead>
<tr>
<th>0.2</th>
<th>0.5</th>
<th>0.9</th>
<th>0.4</th>
<th>0.1</th>
<th>0.6</th>
<th>0.8</th>
<th>0.7</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Fig. 5. An example of sequence vector in the GA

4.2.2 GA operators

As mentioned before, the next generation of the population is generated with respect to crossover and mutation operators.

Crossover: The genetic combination of two or more chromosomes to produce the children for the next generation is called crossover operation. In the present research, three types of the crossover are used: single-point crossover, uniform crossover, and double-point crossover. Figure 6 illustrates the aforementioned crossover.
Mutation: In nature with a small possibility, a child has a characteristic that is not a genetic characteristic of his/her parents. This feature is considered as a mutation in the GA algorithm. In this research, one of the parent’s genes is chosen randomly, and the value of that gene is randomly changed.

**Before mutation**

<table>
<thead>
<tr>
<th>Cut point</th>
<th>Parent chromosome</th>
<th>Cut point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63 0.15 0.25 0.78 0.96 0.12 0.27 0.33 0.74 0.29</td>
<td></td>
<td>0.54 0.26 0.92 0.11 0.68 0.43 0.35 0.31 0.15 0.99</td>
</tr>
</tbody>
</table>

**After mutation**

<table>
<thead>
<tr>
<th>Cut point</th>
<th>Parent chromosome</th>
<th>Cut point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63 0.15 0.25 0.78 0.96 0.12 0.35 0.31 0.15 0.99</td>
<td></td>
<td>0.54 0.26 0.92 0.11 0.68 0.43 0.35 0.31 0.15 0.99</td>
</tr>
</tbody>
</table>

**Fig. 6. Different crossover operators: (a) Single-point crossover; (b) Double-point crossover; (c) Uniform crossover**

**Before mutation**

<table>
<thead>
<tr>
<th>Cut points</th>
<th>Parent chromosome</th>
<th>Cut points</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63 0.15 0.25 0.78 0.96 0.12 0.27 0.33 0.74 0.29</td>
<td></td>
<td>0.54 0.26 0.92 0.11 0.68 0.43 0.35 0.31 0.15 0.99</td>
</tr>
</tbody>
</table>

**Child chromosome**

<table>
<thead>
<tr>
<th>Mask chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0 0 1 0 1 1 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cut points</th>
<th>Parent chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63 0.15 0.25 0.78 0.96 0.12 0.27 0.33 0.74 0.29</td>
<td>0.54 0.26 0.92 0.11 0.68 0.43 0.35 0.31 0.15 0.99</td>
</tr>
</tbody>
</table>

**Fig. 7. An example of mutation operator**

<table>
<thead>
<tr>
<th>Before mutation</th>
<th>After mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63 0.15 0.25 0.78 0.96 0.12 0.27 0.33 0.74 0.29</td>
<td>0.63 0.15 0.25 0.78 0.96 0.47 0.27 0.33 0.74 0.29</td>
</tr>
</tbody>
</table>
4.3. Particle Swarm Optimization (PSO)

PSO algorithm is a population-based algorithm which is firstly introduced by Eberhart and Kennedy in 1995. It coincides with the simulation of the social behavior of the animals such as birds, bees, and fish. In order to simulate the search for the food, the members determine their speed based on two factors: their own best experience and population best experience. Each member in the group (particle) is defined by position and velocity so that the particle position represents the solution for the optimization problem and the velocity shows the direction and distance to guide the movements of the particles. Each particle determines its movement based on its own experience and its neighbors to find the optimal/near optimal solution. Each particle also updates the velocity with respect to the current velocity, the best position by the particle (pbest), and the best position experienced by all particles (gbest). In each iteration, the particle velocity and position are updated as follows:

\[
Vel_i(k+1) = w \times Vel_i(k) + c_1 \times r_1 \times (pbest_i - X_i(k)) + c_2 \times r_2 \times (gbest - X_i(k))
\]

\[
X_i(k+1) = X_i(k) + Vel_i(k+1)
\]

In which, \(Vel_i(k)\) is the velocity of particle \(i\) in iteration \(k\) and \(X_i(k)\) indicates position of particle \(i\) in iteration \(k\). \(pbest_i\) is the best position of particle \(i\) and \(gbest\) is the best position ever obtained. \(r_1\) and \(r_2\) are random numbers between (0,1), \(c_1\) and \(c_2\) are perceptual and social factors, respectively, and \(w\) is the inertial factor.

4.3.1 Solution structure

The solution structure is used to represent the solution similar to the one which is introduced for the SA algorithm.

4.4. Hybrid PSO-SA algorithm

This section is devoted to introduce a hybrid version of the SA and PSO algorithms to solve the FFS problem with outsourcing options. In this approach, the initial population is produced regarding the proposed method which is introduced in the last section. The Boltzmann operator in the SA algorithm is also applied to update the pbest. If the objective function of the new solution is better than the last pbest, the pbest is updated, otherwise the new solution will be accepted as new pbest regarding the obtained probability by the Boltzmann probability function. Other details of this algorithm are the same as the PSO algorithm.

4.5. Calculation the objective function

In all the metaheuristic algorithms, which are explained in previous sections, the objective function is calculated as follows:

First of all, the initial sequence of the jobs is generated regarding sequence vector. Then, entire jobs are processed in the in-house production line. The assigning process to the machine in each stage is as follows. Each job is temporarily assigned to all the available and unavailable machines in each stage. This may be due to the fact that, regarding the unrelated parallel machines at each stage, an unavailable but more efficient machine may produce an earlier completion time for the job. Then, a machine with minimum completion time is selected and the job is assigned to this machine, permanently. This approach is continued until entire jobs are assigned to the machine in each stage. Then the objective function, including tardiness cost and in-house production cost, is calculated.

In the next step, the outsourced jobs must be determined. The last job in the sequence vector is removed from the in-house production line and assigned to any subcontractor. Then, the objective function is approximately estimated. It is due this fact that deleting a job from the in-house production line and assigning it to the subcontractor may lead to changing the tardiness cost of the other jobs. Besides, exact calculation of the objective function for any sequence is very time-consuming. Thus, the approximate estimation of the objective function is taken into account in this stage. If assigning the job to the subcontractor is led to the decreasing the objective value, the exact objective value is calculated and the job is permanently assigned to the corresponding subcontractor. This process is repeated for entire jobs in the sequence. Eventually, tardiness cost, in-house production costs, and outsourced production costs are determined regarding to the subsets of the in-house and outsourced jobs.

5. Experimental Results

This section is devoted to investigating the performance of the proposed approaches for the flexible flow shop scheduling problem with outsourcing options. First of all, some experiments are conducted to tune the metaheuristics parameters. Then, some random test problems are generated in different sizes to analyze the performance of the metaheuristic approaches.

5.1. Parameters Setting

It is obvious that the performance of the metaheuristic algorithms depends on its parameters. As result, we applied Taguchi method to tune the parameters. The levels of the parameters for all the metaheuristic algorithms and their description, legend, and values, are provided in Table 2. For example, four parameters, including maximum of iterations (MaxIt), maximum of sub-iteration (MaxSubIt), initial temperature (T), and temperature reduction rate (alpha) are considered for the simulated annealing algorithm with three levels.
Fig. 8. The signal to noise graph for the SA algorithm

Fig. 9. The signal to noise graph for the GA algorithm
Fig. 10. The signal to noise graph for the PSO algorithm

Fig. 11. The signal to noise graph for the PSO-SA algorithm
As results, the best values of the parameters are summarized in Table 3.

Table 2
The best value of the parameters for the metaheuristic algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>MaxIt</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>MaxSubIt</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>Alpha</td>
<td>0.8</td>
</tr>
<tr>
<td>GA</td>
<td>MaxIt</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Popsizes</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Cross Rate</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>Mut Rate</td>
<td>0.1</td>
</tr>
<tr>
<td>PSO</td>
<td>MaxIt</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Swarm-Size</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.85</td>
</tr>
<tr>
<td>Hybrid PSO-SA</td>
<td>MaxIt</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Swarm-size</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>Alpha</td>
<td>0.7</td>
</tr>
</tbody>
</table>

5.2. Numerical experiments and results for the small and large-size test problems

This section analyzes the performance of the proposed metaheuristic algorithms for the FFS scheduling problem with outsourcing options. For this purpose, two series of the random test problems, small-size test problem and medium to large-size test problems, are generated.

For small size test problems, 25 test problems with respect to Table 4 are generated and the obtained results by the SA, GA, PSO and PSO-SA algorithms are compared with the optimal result.

Table 3
Parameters of the small size test problems

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range of parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of jobs</td>
<td>[4-12]</td>
</tr>
<tr>
<td>Number of stages</td>
<td>[2-10]</td>
</tr>
<tr>
<td>Number of subcontractors</td>
<td>[1-4]</td>
</tr>
<tr>
<td>Number of machines in each stage</td>
<td>[1-4]</td>
</tr>
<tr>
<td>In-house processing time</td>
<td>Uniform (10,30)</td>
</tr>
<tr>
<td>Contractor processing time</td>
<td>Uniform (50,100)</td>
</tr>
<tr>
<td>Transportation time</td>
<td>Uniform (20,50)</td>
</tr>
</tbody>
</table>

The optimal gap (OG) has been used to evaluate the proposed algorithms for the small-size test problems through expression (19):

\[
OG = \frac{\text{Metaheuristic Obj} - \text{Optimal Obj}}{\text{Optimal Obj}}
\]  

(19)

Where, Metaheuristic Obj and Optimal Obj are the optimum value and the objective value obtained by each metaheuristic algorithm, respectively. The experimental results are illustrated in Table 5. It is necessary to mention that the Lingo solver is interrupted after 7200 seconds.
As can be seen in Table 5, the Lingo solver cannot attain the optimal solution for seven test problems. For the other test problems and based on the average OG (\(\bar{O}\)) criteria, the GA (1.4%), PSO (1.4%), PSO-SA (1.6%), and SA (1.6%) have the best performance, respectively. Besides, Regarding the CPU time, the GA and the PSO-SA are the best and the worst algorithms, respectively. Also, the GA can achieve the optimal solution for 5 out of 18 test problems. As results, we can conclude that all the metaheuristic algorithms have good performance to achieve the optimal/near optimal solutions but the GA has the better performance.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of jobs</td>
<td>(20,50,100)</td>
</tr>
<tr>
<td>Number of stages</td>
<td>(5,7,9)</td>
</tr>
<tr>
<td>Number of subcontractors</td>
<td>(3,4,5)</td>
</tr>
<tr>
<td>Number of machines in each stage</td>
<td>(3,5,7)</td>
</tr>
<tr>
<td>In-house Processing time</td>
<td>Uniform (10,30)</td>
</tr>
<tr>
<td>Subcontractor processing time</td>
<td>Uniform (50,100)</td>
</tr>
<tr>
<td>Transportation time</td>
<td>Uniform (20,50)</td>
</tr>
</tbody>
</table>

To solve test problems, the Matlab 2016 software runs on a system with AMD A8-7100 Radeon R5 1.8 GHz processor and 4GB of RAM. The percentage relative error (PTE) has been applied to evaluate the performance of the proposed metaheuristic algorithms.

\[
RPE = \frac{\text{Metaheuristic \ Obj} - \text{Best Obj}}{\text{Best Obj}}
\] (20)

In which, Metaheuristic \ Obj represents the objective function of each metaheuristic algorithm and Best Obj shows the best objective function which is generated by the metaheuristic approaches. The obtained results are presented in Table 7.
### Table 6
The performance of the proposed algorithms for large-size test problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of jobs</th>
<th>Number of stages</th>
<th>Number of subcontractors</th>
<th>Objective function</th>
<th>RPE</th>
<th>CPU</th>
<th>Objective function</th>
<th>RPE</th>
<th>CPU</th>
<th>Objective function</th>
<th>RPE</th>
<th>CPU</th>
<th>Objective function</th>
<th>RPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>50162</td>
<td>1.7%</td>
<td>20</td>
<td>49335</td>
<td>0.0%</td>
<td>26</td>
<td>50663</td>
<td>2.7%</td>
<td>26</td>
<td>50102</td>
<td>1.6%</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>5</td>
<td>4</td>
<td>338365</td>
<td>0.8%</td>
<td>150</td>
<td>335639</td>
<td>0.0%</td>
<td>162</td>
<td>33620</td>
<td>0.2%</td>
<td>142</td>
<td>346608</td>
<td>3.3%</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>111253</td>
<td>0.9%</td>
<td>303</td>
<td>110302</td>
<td>0.0%</td>
<td>345</td>
<td>112847</td>
<td>2.3%</td>
<td>356</td>
<td>110284</td>
<td>0.0%</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>7</td>
<td>3</td>
<td>12498</td>
<td>4.6%</td>
<td>73</td>
<td>12350</td>
<td>3.3%</td>
<td>65</td>
<td>11954</td>
<td>0.0%</td>
<td>68</td>
<td>12233</td>
<td>2.3%</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>7</td>
<td>4</td>
<td>50010</td>
<td>1.3%</td>
<td>211</td>
<td>49364</td>
<td>0.0%</td>
<td>195</td>
<td>50140</td>
<td>1.6%</td>
<td>225</td>
<td>49685</td>
<td>0.7%</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>7</td>
<td>5</td>
<td>120054</td>
<td>1.3%</td>
<td>549</td>
<td>118987</td>
<td>0.4%</td>
<td>558</td>
<td>118478</td>
<td>0.0%</td>
<td>554</td>
<td>119548</td>
<td>0.9%</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>9</td>
<td>3</td>
<td>20811</td>
<td>0.8%</td>
<td>211</td>
<td>20032</td>
<td>0.0%</td>
<td>265</td>
<td>20254</td>
<td>1.1%</td>
<td>224</td>
<td>20521</td>
<td>2.4%</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>9</td>
<td>4</td>
<td>58012</td>
<td>3.9%</td>
<td>433</td>
<td>57564</td>
<td>3.4%</td>
<td>425</td>
<td>57819</td>
<td>0.4%</td>
<td>452</td>
<td>58124</td>
<td>1.0%</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>9</td>
<td>5</td>
<td>98829</td>
<td>3.0%</td>
<td>835</td>
<td>99245</td>
<td>3.4%</td>
<td>884</td>
<td>95987</td>
<td>0.0%</td>
<td>846</td>
<td>96875</td>
<td>0.9%</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>17005</td>
<td>0.7%</td>
<td>27</td>
<td>16879</td>
<td>0.0%</td>
<td>30</td>
<td>17548</td>
<td>4.0%</td>
<td>25</td>
<td>16985</td>
<td>0.6%</td>
</tr>
<tr>
<td>11</td>
<td>50</td>
<td>5</td>
<td>4</td>
<td>36698</td>
<td>2.8%</td>
<td>184</td>
<td>36945</td>
<td>3.5%</td>
<td>192</td>
<td>35689</td>
<td>0.0%</td>
<td>235</td>
<td>36258</td>
<td>1.6%</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>92031</td>
<td>2.2%</td>
<td>340</td>
<td>90025</td>
<td>0.0%</td>
<td>356</td>
<td>90215</td>
<td>0.2%</td>
<td>345</td>
<td>91548</td>
<td>1.7%</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>7</td>
<td>3</td>
<td>21530</td>
<td>2.6%</td>
<td>98</td>
<td>21458</td>
<td>2.2%</td>
<td>86</td>
<td>21085</td>
<td>0.5%</td>
<td>98</td>
<td>20987</td>
<td>0.0%</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>7</td>
<td>4</td>
<td>51326</td>
<td>2.8%</td>
<td>256</td>
<td>49919</td>
<td>0.0%</td>
<td>298</td>
<td>50154</td>
<td>0.5%</td>
<td>245</td>
<td>51045</td>
<td>2.3%</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>7</td>
<td>5</td>
<td>141645</td>
<td>1.0%</td>
<td>531</td>
<td>142359</td>
<td>1.5%</td>
<td>536</td>
<td>140254</td>
<td>0.0%</td>
<td>530</td>
<td>142569</td>
<td>1.7%</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>9</td>
<td>3</td>
<td>20901</td>
<td>2.7%</td>
<td>207</td>
<td>20354</td>
<td>0.0%</td>
<td>235</td>
<td>21045</td>
<td>3.4%</td>
<td>204</td>
<td>20860</td>
<td>2.5%</td>
</tr>
<tr>
<td>17</td>
<td>50</td>
<td>9</td>
<td>4</td>
<td>51194</td>
<td>1.4%</td>
<td>435</td>
<td>50487</td>
<td>0.0%</td>
<td>436</td>
<td>52415</td>
<td>3.8%</td>
<td>432</td>
<td>50789</td>
<td>0.6%</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>9</td>
<td>5</td>
<td>116784</td>
<td>1.1%</td>
<td>898</td>
<td>116987</td>
<td>1.3%</td>
<td>885</td>
<td>115468</td>
<td>0.0%</td>
<td>904</td>
<td>117845</td>
<td>2.1%</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>13685</td>
<td>3.3%</td>
<td>22</td>
<td>13320</td>
<td>0.6%</td>
<td>19</td>
<td>13824</td>
<td>0.0%</td>
<td>23</td>
<td>13478</td>
<td>1.8%</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>5</td>
<td>4</td>
<td>44586</td>
<td>2.3%</td>
<td>145</td>
<td>43568</td>
<td>0.0%</td>
<td>156</td>
<td>45678</td>
<td>4.8%</td>
<td>184</td>
<td>44687</td>
<td>2.6%</td>
</tr>
<tr>
<td>21</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>87571</td>
<td>1.0%</td>
<td>386</td>
<td>88547</td>
<td>1.1%</td>
<td>394</td>
<td>89654</td>
<td>2.4%</td>
<td>325</td>
<td>88546</td>
<td>1.1%</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>7</td>
<td>3</td>
<td>17144</td>
<td>0.8%</td>
<td>165</td>
<td>17016</td>
<td>0.0%</td>
<td>158</td>
<td>17854</td>
<td>4.9%</td>
<td>135</td>
<td>17175</td>
<td>0.9%</td>
</tr>
<tr>
<td>23</td>
<td>50</td>
<td>7</td>
<td>4</td>
<td>44511</td>
<td>2.1%</td>
<td>258</td>
<td>43651</td>
<td>0.2%</td>
<td>221</td>
<td>44884</td>
<td>3.0%</td>
<td>221</td>
<td>43578</td>
<td>0.0%</td>
</tr>
<tr>
<td>24</td>
<td>100</td>
<td>7</td>
<td>5</td>
<td>97550</td>
<td>2.6%</td>
<td>538</td>
<td>95119</td>
<td>0.0%</td>
<td>584</td>
<td>98546</td>
<td>3.6%</td>
<td>589</td>
<td>95487</td>
<td>0.4%</td>
</tr>
<tr>
<td>25</td>
<td>20</td>
<td>9</td>
<td>3</td>
<td>19002</td>
<td>1.3%</td>
<td>225</td>
<td>18754</td>
<td>0.0%</td>
<td>224</td>
<td>18978</td>
<td>1.0%</td>
<td>212</td>
<td>18765</td>
<td>0.1%</td>
</tr>
<tr>
<td>26</td>
<td>50</td>
<td>9</td>
<td>4</td>
<td>58685</td>
<td>0.0%</td>
<td>585</td>
<td>59876</td>
<td>2.0%</td>
<td>586</td>
<td>60124</td>
<td>2.5%</td>
<td>545</td>
<td>59876</td>
<td>2.0%</td>
</tr>
<tr>
<td>27</td>
<td>100</td>
<td>9</td>
<td>5</td>
<td>148781</td>
<td>2.7%</td>
<td>932</td>
<td>145273</td>
<td>0.3%</td>
<td>975</td>
<td>150245</td>
<td>3.7%</td>
<td>1035</td>
<td>144875</td>
<td>0.0%</td>
</tr>
<tr>
<td>28</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>21152</td>
<td>5.0%</td>
<td>26</td>
<td>20154</td>
<td>0.0%</td>
<td>26</td>
<td>20602</td>
<td>2.2%</td>
<td>28</td>
<td>20481</td>
<td>1.6%</td>
</tr>
<tr>
<td>29</td>
<td>50</td>
<td>5</td>
<td>4</td>
<td>44801</td>
<td>0.0%</td>
<td>114</td>
<td>45214</td>
<td>0.9%</td>
<td>124</td>
<td>45789</td>
<td>2.2%</td>
<td>114</td>
<td>46578</td>
<td>4.0%</td>
</tr>
</tbody>
</table>

82
| 30 | 100 | 5 | 5 | 122195 | 2.3% | 318 | 119457 | 0.0% | 300 | 123546 | 3.4% | 304 | 124587 | 4.3% |
| 31 | 20 | 7 | 3 | 14927 | 2.5% | 75 | 14568 | 0.0% | 75 | 14803 | 1.6% | 78 | 15076 | 3.5% |
| 32 | 50 | 7 | 4 | 56564 | 3.7% | 255 | 55486 | 1.7% | 256 | 55208 | 1.2% | 278 | 54568 | 0.0% |
| 33 | 100 | 7 | 5 | 114829 | 0.2% | 438 | 115452 | 0.8% | 414 | 115632 | 0.9% | 470 | 114587 | 0.0% |
| 34 | 20 | 9 | 3 | 18236 | 2.2% | 235 | 17845 | 0.0% | 215 | 18141 | 1.7% | 235 | 18440 | 3.3% |
| 35 | 50 | 9 | 4 | 46863 | 0.0% | 542 | 46937 | 0.2% | 580 | 47421 | 1.2% | 660 | 46987 | 0.3% |
| 36 | 100 | 9 | 5 | 110639 | 0.9% | 838 | 109684 | 0.0% | 808 | 112544 | 2.6% | 896 | 110633 | 0.9% |
| 37 | 20 | 5 | 3 | 12342 | 1.5% | 14 | 12157 | 0.0% | 12 | 12546 | 3.2% | 14 | 12754 | 4.9% |
| 38 | 50 | 5 | 4 | 35346 | 0.0% | 186 | 35506 | 0.5% | 187 | 36231 | 2.5% | 184 | 36108 | 2.2% |
| 39 | 100 | 5 | 5 | 89420 | 1.1% | 312 | 88927 | 0.5% | 302 | 88451 | 0.0% | 351 | 91045 | 2.9% |
| 40 | 20 | 7 | 3 | 25110 | 2.4% | 87 | 24519 | 0.0% | 84 | 25689 | 4.8% | 78 | 24655 | 0.6% |
| 41 | 50 | 7 | 4 | 64923 | 7.8% | 250 | 60215 | 0.0% | 270 | 61475 | 2.1% | 258 | 63542 | 5.5% |
| 42 | 100 | 7 | 5 | 158722 | 2.9% | 532 | 154258 | 0.0% | 565 | 154789 | 0.3% | 521 | 155421 | 0.8% |
| 43 | 20 | 9 | 3 | 19393 | 4.0% | 189 | 18845 | 1.1% | 198 | 18645 | 0.0% | 178 | 19275 | 3.4% |
| 44 | 50 | 9 | 4 | 47419 | 3.4% | 447 | 45876 | 0.0% | 456 | 47956 | 4.5% | 495 | 48654 | 6.1% |
| 45 | 100 | 9 | 5 | 120546 | 0.8% | 859 | 121457 | 1.6% | 884 | 119548 | 0.0% | 909 | 120124 | 0.5% |
| 46 | 20 | 5 | 3 | 17327 | 0.7% | 12 | 17209 | 0.0% | 14 | 17854 | 3.7% | 11 | 17744 | 3.1% |
| 47 | 50 | 5 | 4 | 39411 | 3.7% | 183 | 38956 | 2.5% | 176 | 38002 | 0.0% | 195 | 40524 | 6.6% |
| 48 | 100 | 5 | 5 | 102458 | 2.3% | 414 | 100175 | 0.0% | 425 | 101369 | 1.2% | 414 | 100361 | 0.2% |
| 49 | 20 | 7 | 3 | 15802 | 2.0% | 220 | 15485 | 0.0% | 214 | 16142 | 4.2% | 235 | 15831 | 2.2% |
| 50 | 50 | 7 | 4 | 41996 | 1.1% | 254 | 41687 | 0.4% | 264 | 42755 | 2.9% | 242 | 41534 | 0.0% |
| 51 | 100 | 7 | 5 | 88956 | 1.7% | 553 | 87456 | 0.0% | 565 | 89542 | 2.4% | 575 | 88654 | 1.4% |
| 52 | 20 | 9 | 3 | 22457 | 4.7% | 264 | 21657 | 0.0% | 265 | 21743 | 1.3% | 270 | 22146 | 3.2% |
| 53 | 50 | 9 | 4 | 56875 | 2.5% | 592 | 55478 | 0.0% | 504 | 57865 | 4.3% | 590 | 56478 | 1.8% |
| 54 | 100 | 9 | 5 | 165111 | 0.0% | 915 | 165724 | 0.4% | 951 | 166458 | 0.8% | 955 | 169487 | 2.7% |
| 55 | 20 | 5 | 3 | 19012 | 3.7% | 22 | 18338 | 0.0% | 24 | 19208 | 4.7% | 19 | 18457 | 0.6% |
| 56 | 50 | 5 | 4 | 52836 | 2.7% | 123 | 51427 | 0.0% | 162 | 51633 | 0.4% | 138 | 52411 | 1.9% |
| 57 | 100 | 5 | 5 | 153687 | 0.0% | 309 | 154215 | 0.3% | 345 | 155478 | 1.2% | 315 | 154215 | 0.3% |
| 58 | 20 | 7 | 3 | 17456 | 3.4% | 135 | 16989 | 0.7% | 124 | 16875 | 0.0% | 132 | 17001 | 0.7% |
| 59 | 50 | 7 | 4 | 43875 | 4.1% | 244 | 42154 | 0.0% | 248 | 43548 | 3.3% | 264 | 44123 | 4.7% |
| 60 | 100 | 7 | 5 | 120992 | 0.9% | 514 | 119875 | 0.0% | 585 | 121471 | 1.3% | 574 | 125486 | 4.7% |
| 61 | 20 | 9 | 3 | 222461 | % | 212 | 22145 | % | 204 | 21545 | % | 240 | 22314 | % | 3.6 | 240 |
| 62 | 50 | 9 | 4 | 46174 | 2.0 | 418 | 45285 | 0.0 | 456 | 46532 | 2.8 | 0.0 | 435 | 45875 | 1.3 | 452 |
| 63 | 100 | 9 | 5 | 109886 | 0.9 | 848 | 110245 | 1.3 | 887 | 108856 | 0.0 | 924 | 109206 | 2.1 | 945 |
| 64 | 20 | 5 | 3 | 16194 | 4.1 | 128 | 39827 | 0.4 | 132 | 39678 | 0.0 | 19 | 15648 | 1.1 | 14 |
| 65 | 50 | 5 | 4 | 41318 | 4.1 | 374 | 84752 | 2.7 | 358 | 86598 | 4.9 | 395 | 87542 | 6.1 | 412 |
| 66 | 100 | 5 | 5 | 82547 | 0.0 | 139 | 17739 | 0.0 | 142 | 18235 | 2.8 | 136 | 18088 | 2.0 | 139 |
| 67 | 20 | 7 | 3 | 18492 | 4.2 | 276 | 56879 | 0.0 | 306 | 58689 | 3.2 | 297 | 57456 | 1.0 | 258 |
| 68 | 50 | 7 | 4 | 59343 | 4.3 | 565 | 148963 | 0.3 | 556 | 150214 | 1.1 | 587 | 148562 | 0.0 | 535 |
| 69 | 100 | 7 | 5 | 152487 | 2.6 | 228 | 18756 | 0.0 | 225 | 18754 | 0.0 | 226 | 18956 | 1.1 | 225 |
| 70 | 20 | 9 | 3 | 19443 | 3.7 | 474 | 64510 | 0.4 | 468 | 64235 | 0.0 | 474 | 68754 | 0.0 | 465 |
| 71 | 50 | 9 | 4 | 65333 | 1.7 | 938 | 120124 | 0.0 | 945 | 122254 | 1.8 | 954 | 123254 | 2.6 | 965 |
| 72 | 100 | 9 | 5 | 121454 | 1.1 | 13 | 14568 | 0.0 | 14 | 14572 | 0.0 | 14 | 15342 | 5.3 | 16 |
| 73 | 20 | 5 | 3 | 15185 | 4.2 | 184 | 36584 | 0.0 | 198 | 37684 | 3.0 | 186 | 37568 | 2.7 | 196 |
| 74 | 50 | 5 | 4 | 38654 | 5.7 | 345 | 101245 | 1.0 | 332 | 103548 | 3.3 | 332 | 100245 | 0.0 | 324 |
| 75 | 100 | 5 | 5 | 106163 | 5.9 | 123 | 13873 | 0.0 | 124 | 14256 | 2.8 | 121 | 14668 | 5.7 | 125 |
| 76 | 20 | 7 | 3 | 14025 | 1.1 | 255 | 40125 | 1.4 | 252 | 42154 | 6.5 | 268 | 40387 | 2.0 | 286 |
| 77 | 50 | 7 | 4 | 39587 | 0.0 | 510 | 106866 | 0.3 | 532 | 106582 | 0.0 | 563 | 110240 | 3.4 | 614 |
| 78 | 100 | 7 | 5 | 108654 | 1.9 | 242 | 22047 | 0.0 | 248 | 22455 | 1.9 | 242 | 22456 | 2.4 | 245 |
| 79 | 20 | 9 | 3 | 22455 | 1.9 | 425 | 62531 | 0.0 | 435 | 64578 | 3.3 | 442 | 64057 | 0.2 | 458 |
| 80 | 50 | 9 | 4 | 63969 | 2.3 | 986 | 143024 | 0.0 | 925 | 149685 | 4.7 | 954 | 143257 | 0.0 | 976 |
| 81 | 100 | 9 | 5 | 148472 | 3.8 | 28 | 16354 | 0.0 | 31 | 17240 | 5.4 | 27 | 16587 | 1.4 | 25 |
| 82 | 20 | 5 | 3 | 17100 | 4.6 | 152 | 43621 | 0.1 | 150 | 43578 | 0.0 | 168 | 44228 | 1.5 | 136 |
| 83 | 50 | 5 | 4 | 44245 | 1.5 | 370 | 128803 | 0.0 | 370 | 129867 | 0.8 | 396 | 130454 | 1.3 | 404 |
| 84 | 100 | 5 | 5 | 133645 | 3.8 | 94 | 16476 | 1.1 | 92 | 16584 | 1.7 | 96 | 17104 | 4.9 | 110 |
| 85 | 20 | 7 | 3 | 16304 | 0.0 | 247 | 50254 | 0.0 | 253 | 53212 | 5.9 | 275 | 51427 | 2.3 | 265 |
| 86 | 50 | 7 | 4 | 54114 | 7.7 | 532 | 110422 | 4.7 | 568 | 108666 | 3.0 | 654 | 105487 | 0.0 | 602 |
| 87 | 100 | 7 | 5 | 109655 | 4.0 | 172 | 15027 | 0.0 | 166 | 15784 | 5.0 | 196 | 15227 | 1.3 | 196 |
| 88 | 20 | 9 | 3 | 15247 | 1.5 | 412 | 44124 | 0.2 | 425 | 45732 | 3.8 | 412 | 44055 | 0.0 | 435 |
| 89 | 50 | 9 | 4 | 46875 | 6.4 | 876 | 120704 | 0.0 | 902 | 121042 | 0.3 | 956 | 122335 | 1.4 | 967 |
| 90 | 100 | 9 | 5 | 122179 | 1.2 | 17 | 11200 | 0.0 | 12 | 11447 | 2.2 | 16 | 11347 | 1.3 | 19 |
92 50 5 4 39127 1.2 % 141 38654 0.0 % 123 41247 6.7 % 154 39754 2.8 % 112
93 100 5 5 91300 0.8 % 364 90993 0.5 % 358 90554 0.0 % 348 93655 3.4 % 386
94 20 7 3 24580 3.7 % 126 23713 0.0 % 140 25477 7.4 % 128 24875 4.9 % 132
95 50 7 4 57899 5.7 % 188 54755 0.0 % 192 58668 7.1 % 196 57854 5.7 % 192
96 100 7 5 118465 0.0 % 577 121457 2.5 % 556 119586 0.9 % 525 123547 4.3 % 572
97 20 9 3 14131 4.3 % 236 13548 0.0 % 225 14102 4.1 % 222 13547 0.0 % 230
98 50 9 4 50036 0.7 % 469 49868 0.4 % 458 51457 3.6 % 475 49669 0.0 % 468
99 100 9 5 142154 1.4 % 948 140257 0.0 % 932 144755 3.2 % 954 146587 4.5 % 969
100 20 5 5 13365 7.4 % 20 12445 0.0 % 22 12668 1.8 % 19 12547 0.8 % 19

Average

Regarding Table 7, the GA algorithm outperforms the other algorithms with respect to the average RPE. Furthermore, there is a significant difference between the efficiency of the algorithms. Also, the GA can achieve the best solution among the algorithms for 56 out of 100 test problems.

In order to verify the statistical validity of the results shown in Table 7 and to confirm which algorithm has better performance, a two-sample t-test is conducted in 95% confidence interval. Note that the statistical hypothesis, which is considered in this research, is as: \( H_0: \mu_1 - \mu_2 = 0 \) \( \text{vs.} \) \( H_1: \mu_1 - \mu_2 \neq 0 \). The obtained results are summarized in Table 8.

### Table 7
**P-Value of the two sample t-test for the proposed algorithms**

<table>
<thead>
<tr>
<th></th>
<th>SA</th>
<th>GA</th>
<th>PSO</th>
<th>PSO-SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>---</td>
<td>0.000</td>
<td>0.037</td>
<td>0.146</td>
</tr>
<tr>
<td>GA</td>
<td>---</td>
<td>---</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>PSO</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>0.605</td>
</tr>
<tr>
<td>PSO-SA</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

Regarding Table 8, the GA outperforms other proposed algorithms with respect to the \( P-Value < 0.05 \). Also, other algorithms have the same performance in the confidence interval of 0.95. The means plot and least significant difference (LSD) intervals (at the 95 % confidence level) are depicted in Figure 12 for four algorithms.

![Fig. 12. The LSD plot of the proposed algorithms](image-url)
5.3. Analysis of parameters of test problems

Analysis of the number of jobs: in order to investigate the effect of number of jobs on the proposed algorithms, the interaction between the algorithms and number of jobs is illustrated in Figure 13. As can be seen, in the entire cases, the GA has better performance.

![Fig. 13. Plot of average RPE for the interaction between the algorithms and number of jobs](image1)

Analysis of the number of stages: another plot for interaction between the algorithms and number of stages is depicted in Figure 14. Considering Figure 14, the GA works better than other algorithms in all cases.

![Fig. 14. Plot of average RPE for the interaction between the algorithms and number of stages](image2)

Analysis of the number of subcontractors: finally, Figure 15 shows the interaction plot between number of subcontractors and quality of the algorithms. Regarding to Figure 15, the GA works better than the PSO, SA, and PSO-SA.

![Fig. 15. Plot of average RPE for the interaction between the algorithms and number of subcontractors](image3)
6. Conclusion

Flexible Flow Shop is a common manufacturing system in which a set of $n$ jobs are processed on $s$ stages. This paper aims to consider outsourcing options in a flexible flow shop scheduling problem with cost-related objective functions. In this paper, we developed a mathematical model for the research problem. Regarding the NP-hardness of the research problem, we used metaheuristic algorithms, SA, GA, PSO and PSO-SA for solving problems with medium to large-size test problems. The obtained results demonstrated that the GA has better performance. Future research may consider outsourcing for other scheduling environments or consider some assumptions such as release date for jobs or consider the batch shipment constraint for the outsourced jobs. Proposing other heuristic and metaheuristic approaches, considering other objective functions, and using data from the real case study can be other clues of future researches.

Acknowledgment

The second author would like to thank the financial support of the Babol Noshirvani University of Technology for this research under grant number BNUT/388026/98.

References


http://www.qjie.ir/article_677014.html
DOI:10.22094/JOIE.2020.1873983.1674