International Journal of Industrial Engineering & Production Research (2019) 30: 207-223
DOI: 10.22068/ijiepr.30.2.207

RESEARCH PAPER

A Simultaneous Worker Assignment and Scheduling Problem for Minimizing Makespan in Flexible Flow Shop Via Metaheuristic Approaches

Fatemeh Bozorgnezhad[1], Ebrahim Asadi-Gangraj[*2] & Mohammad Mahdi Paydar[3]

Received 05 Novebmer 2018; Revised 28 April 2019; Accepted 8 May 2019; Publish online 20 June 2019

ABSTRACT
In many real scheduling situations, it is necessary to deal with the worker assignment and job scheduling together. However, in traditional scheduling problems, only the machine is assumed to be a constraint, and there is not any constraint about workers. This assumption could, in part, be due to the lower cost of workers compared to machines or the complexity of workers' assignment problems. This research proposes a flexible flow shop scheduling problem with two simultaneous issues: finding the best worker assignment and solving the corresponding scheduling problem. A mathematical model that extends a flexible flow shop scheduling problem is presented to admit the worker assignment. Due to the NP-hardness of the research problem, two approximation approaches based on particle swarm optimization, named PSO and SPSO, are applied to minimize the makespan. The experimental results show that the proposed algorithms can efficiently minimize the makespan; however, the SPSO generates better solutions, especially for large-sized problems.

KEYWORDS *Flexible flow shop, Worker assignment, MILP model, Particle swarm optimization, Simulated annealing.*

1. Introduction

Scheduling algorithms and models are most widely used in manufacturing systems for efficient production and management. After the first study in 1971 by Arthanari and Ramamurthy [1], flexible flow shop (FFS) scheduling problem has been a popular and noticeable research problem among the researchers due to its practical and theoretical significance. It is also widely used in real industries such as the automotive industry, chemical industry, metallurgical industry, and iron manufacturing [2]. The FFS is a production system that consists of a set of two or more stages with at least one stage having two or more parallel (related or unrelated) machines [3]. It is also called hybrid flow shop, flexible flow line or flow shop with multiple processors [4]. A job in such a system consists of a sequence of operations processed in successive stages, and all jobs pass through processing stages in the same order. In a stage with parallel machines, a job can be processed on any of the parallel machines.

Most scheduling approaches assume that the processing time of every job in each stage is independent of the worker that performs it. It can cause serious problems in the realistic scheduling problem. On the other hand, in real-world manufacturing systems, instead of machine scheduling, assignment of the worker to each machine at each stage is a critical issue and can affect the efficiency of the scheduling problems. Therefore, the worker assignment and job scheduling decisions need to be dealt with jointly to generate better solutions. On the other hand, the worker assignment and scheduling problem have been rarely studied in the literature in the FFS environment with unrelated parallel machines, simultaneously. Therefore, in this paper, the extension of a flexible flow shop scheduling problem is proposed, where two simultaneous issues must be considered: first,

* Corresponding author: *Ebrahim Asadi-Gangraj*
e.asadi@nit.ac.ir

1. MSc student of industrial engineering; Babol Noshirvani University of Technology; Babol; Iran.
2. Assistant professor of industrial engineering; Babol Noshirvani University of Technology; Babol; Iran.
3. Assistant professor of industrial engineering; Babol Noshirvani University of Technology; Babol; Iran.

finding the best worker assignment to each machine in each stage and, second, solving the corresponding scheduling problem. The underlying assumption in this research is that the processing time of a job performed on a machine depends upon the assigned worker. This assumption is usually realistic in many industries because the workers have different skill levels to perform a particular job. Thus, for this situation, the completion time of a job is affected by two main decisions: as usual, the jobs sequencing and scheduling and the optimal assignment of the workers to each machine in every stage, which helps to achieve better solutions in the search space.

To the best of our knowledge, the problem of worker assignment and jobs scheduling in the flexible flow shop with the unrelated parallel machine has not been addressed in the literature. This research aims to present a new mixed integer linear programming (MILP) model to solve the problem optimally. Since the production scheduling in an FFS environment is, in most cases, NP-hard [5], the candidate problem is also NP-hard, and the MILP model cannot obtain the optimal solution in an amount of reasonable time; therefore, the application of the metaheuristic approaches to tackle the problem under investigation is inevitable [6]. Therefore, two versions of the particle swarm optimization are presented, named PSO and SPSO, to solve it, approximately. PSO algorithm has attracted researchers' attention during the last years. It is an evolutional algorithm performed on a pool of feasible solutions, called particles. These particles move around in the search space to achieve high-quality solutions. To avoid premature convergence of the PSO, a new hybrid Simulated Annealing-Particle Swarm Optimization, called SPSO, is presented based on the idea that PSO ensures the fast convergence, while SA brings the search out of local optima because of its strong local search ability. The selected objective of the scheduling problem is to minimize the maximum completion times (makespan).

This study is organized as follows: Section 2 reviews the literature on the related scheduling problems. The proposed MILP model and metaheuristic algorithms are introduced in Section 3 to solve the problem. The experimental results are illustrated in Section 4. Conclusion and future research are given in Section 5.

2. A Brief Overview of the Literature Review

This section is included in two sub-sections. First, some works existing in the literature that consider the scheduling problems with the worker assignment are discussed. In the second sub-section, some studies proposed in the literature about flexible flow shop scheduling problem are presented, which applied different metaheuristic approaches.

Behnamian [7] proposed colonial competitive algorithm improved by variable neighborhood search algorithm for the simultaneous effects of learning and deterioration on hybrid flow shop scheduling with sequence-dependent setup times. He assumed that the processing time of any job depends on the number of workers assigned to the job at any stage. The objective function was to minimize the sum of the earliness, tardiness, makespan, and total worker employing costs. Benavides, Ritt and Miralles [8] studied the extension to the flow shop scheduling problem named Heterogeneous Flow Shop Scheduling Problem, where two simultaneous problems must be solved: finding the best worker assignment to each workstation and solving the resulting scheduling problem. They presented a mathematical model that extends a flow shop model for the heterogeneous worker assignment and proposed a heuristic method based on scatter search and path relinking to minimize the makespan.

Celano, Costa and Fichera [9] developed a mathematical model and optimization procedure to find efficient solutions to the flow shop group scheduling and workers assignment problem with sequence-dependent setup time. They proposed a proper genetic algorithm and tested it by running a benchmark with different scenarios such as numbers of machines, groups, jobs, worker skills, and learning ability. Tyagi et al. [10] studied the flexible flow shop scheduling problem with learning and forgetting effect of workers and sequence-dependent setup times to minimize the weighted sum of the maximum completion time and maximum tardiness. The learning effect occurs when a worker's skill increases after repeating similar job, resulting in the processing time decrease.

Hu [11,12,13,14] studied the worker assignment scheduling problem on an identical parallel machine with different objective functions, total tardiness, and total flow time. He applied some heuristic approaches and simulation processes to

solve the problems. Chaudhry and Drake [15] considered worker assignment problem in the identical parallel machines environment with total tardiness minimization. They proposed a spreadsheet-based GA approach to solve the problem, approximately. Chaudhry [16] considered the worker assignment and scheduling problem in parallel machines environment. He proposed a spreadsheet-based domain-independent general-purpose genetic algorithm (GA) to minimize the total flow time.

Carniel, Benavides and Ritt [17] studied the insertion of workers with disabilities into flow shops to minimize the makespan. To solve the problem exactly, they proposed several mathematical models. Aftab, Muhammad and Riaz [18] proposed an Ant Colony Optimization (ACO) approach to minimize the makespan for scheduling the jobs and assigning the workers for uniformly related parallel machines.

Since the FFS problem is proven to be NP-hard, the worker assignment and FFS scheduling problem is obviously NP-hard. Thus, the exact methods are incapable of solving real-world instances on medium to large scales. Thus, it is inevitable to find non-exact algorithms to deal with the problem under investigation. Many researchers applied metaheuristic algorithms to solve such problems with different assumptions and to generate near-optimal solutions with considerably low computational time during the last years [19,20,21,22,23]. The most popular metaheuristic approaches to solving the FFS problems are genetic algorithm (GA), simulated annealing (SA), and tabu search (TS). More recently, particle swarm optimization (PSO) algorithm has attracted researchers' attention such that some PSO procedures can be found for solving the FFS scheduling problems, approximately.

Rajaee Abyaneh and Gholami [24] studied a hybrid flow shop scheduling problem with unrelated parallel machines to minimize the sum of earliness and tardiness. They used GPSO, PSO algorithm combined with genetic operators, to solve this problem. Tadayon and Salmasi [25] considered a flexible flow shop group scheduling problem with makespan minimization and applied some versions of the PSO algorithm to solve it.

Ou, Zou and Gao [26] investigated a flexible flow shop scheduling problem and formulated it as an integer programming. They proposed a hybrid particle swarm optimization (HPSO) to solve the problem. Tang et al. [27] considered dynamically flexible flow shop scheduling to minimize energy consumption and makespan. They proposed a PSO algorithm to search the Pareto optimal solutions. Singh and Mahapatra [28] presented a flexible flow shop scheduling problem and applied a particle swarm optimization algorithm to solve it. Li, Pan and Mao [29] proposed a hybrid PSO algorithm and iterated local search to solve the hybrid flow shop scheduling with preventive maintenance activities. They applied different crossover and mutation operators to enhance the search ability of the proposed algorithm.

For the reader's convenience, some of the main research studies in the context of scheduling problem with worker assignment are summarized in Table 1.

Tab. 1. A brief overview of the literature review

Year	Author(s)	Environment	Objective Function	Solution approach	Description
2014	Behnamian [7]	Hybrid flow shop	Sum of the earliness, tardiness, makespan, and total worker employing costs	Hybrid colonial competitive algorithm	Worker-dependent processing time
2014	Benavides, Ritt and Miralles [8]	Heterogeneous Flow Shop	Makespan	Heuristic method based on scatter search and path relinking	Worker assignment
2011	Celano, Costa and Fichera [9]	Flow shop group scheduling	Makespan	GA	Worker assignment
2014	Tyagi et al. [10]	Parallel machine	Weighted sum of the maximum completion time and maximum tardiness	Heuristic algorithms	learning and forgetting effect of the workers
2004-	Hu [11,12,13,14]	Identical	Total tardiness and total	Heuristic	Worker

2006		parallel machine	flow time	algorithms	assignment
2009	Chaudhry and Drake [15]	Identical parallel machine	total tardiness minimization	Spreadsheet-based GA	Worker assignment
2010	Chaudhry [16]	Identical parallel machine	Total flow time	spreadsheet-based domain independent general purpose GA	Worker assignment
2013	Carniel, Benavides and Ritt [17]	Flow shop	Makespan	MILP models	Workers with disabilities
2012	Aftab, Muhammad and Riaz [18]	Uniformly related parallel machines	Makespan	ACO	Worker assignment
2019	Present research	Flexible flow shop	Makespan	PSO, SPSO	Worker assignment

3. Methodology

3-1. Problem description

There are n different jobs, where each job must be processed by one machine at each stage. The considered production environment is as flexible flow shop, in which there is a group of machines arranged into s stages in series; in stage t, there are S_t unrelated machines in parallel. Further, there are W workers so that each worker must be assigned to one machine at all the stages. The following assumptions are made in this research:

- Entire jobs workers are available at the initial time.
- All m stages are independent.
- Preemption is prohibited in this research.
- There are no buffer capacity constraints between stages.
- One job can be processed only by one machine at any time, and one machine can process only one job at a time.
- The processing sequence is known and based on the assigned worker.
- The setup time of all the jobs is included in the processing time.

3-2. Mathematical formulation

In this section, the research problem is expressed formally as a mixed integer linear programming (MILP) model. As mentioned above, the objective function of the candidate problem is to minimize the makespan. The indices, parameters, decision variables, and the MILP model are given as follows:

Indices

n: Number of jobs ($j, l = 1,2, \dots, n$)

T: Number of stages ($t = 1,2, \dots, T$)

S_t: Number of machines at stage t ($i = 1,2, \dots S_t$)

W: Number of works with different skill levels ($w = 1,2, \dots, W$)

Parameters

P_{ijwt}: Processing times of job j on machine i by worker w in stage t

M: A very large number

Decision variables

C_{max} : Maximum completion time

C_{jt} : Completion time of job j at stage t

P'_{ijt} : Processing time of job j on machine i at stage t

$$X_{ijt} = \begin{cases} 1 & if\ \ job\ j\ processed\ on\ machine\ i\ at\ stage\ t \\ 0 & otherwise \end{cases}$$

$$Y_{iljt} = \begin{cases} 1 & if\ job\ l\ is\ processed\ before\ job\ j\ on\ machine\ i\ at\ stage\ t \\ 0 & otherwise \end{cases}$$

$$Z_{iwt} = \begin{cases} 1 & if\ worker\ w\ is\ assigned\ to\ machine\ i\ at\ stage\ t \\ 0 & otherwise \end{cases}$$

Therefore, the mathematical model of the FFS scheduling problem with worker assignment can be formulated as follows:

$$Z = \min C_{max} \quad (1)$$

$$\sum_{i=1}^{S_t} X_{ijt} = 1 \qquad j = 1,2,\dots,n, t = 1,2,\dots,T \qquad (2)$$

$$C_{j1} \geq \sum_{i=1}^{S_1} P'_{ij1} X_{ij1} \qquad j = 1,2,\dots,n \qquad (3)$$

$$C_{jt} \geq C_{j,t-1} + \sum_{i=1}^{S_t} P'_{ijt} X_{ijt} \qquad j = 1,2,\dots,n, t = 2,\dots,T \qquad (4)$$

$$C_{jt} + M\left(3 - X_{ijt} - X_{ilt} - Y_{iljt}\right) \geq C_{lt} + P'_{ijt} X_{ijt} \qquad j,l = 1,2,\dots,n;\ j \neq l,\ t = 1,2,\dots,T, i = 1,2,\dots,S_t \qquad (5)$$

$$C_{lt} + M\left(2 - X_{ijt} - X_{ilt} + Y_{iljt}\right) \geq C_{jt} + P'_{ilt} X_{ilt} \qquad j,l = 1,2,\dots,n;\ j \neq l,\ t = 1,2,\dots,T, i = 1,2,\dots,S_t \qquad (6)$$

$$P'_{ijt} = \sum_{w=1}^{W} P_{ijwt} Z_{iwt} \qquad j = 1,2,\dots,n,\ t = 1,2,\dots,T, i = 1,2,\dots,S_t \qquad (7)$$

$$\sum_{w=1}^{W} Z_{iwt} = 1 \qquad t = 1,2,\dots,T, i = 1,2,\dots,S_t \qquad (8)$$

$$\sum_{t=1}^{T} \sum_{i=1}^{S_t} Z_{iwt} = 1 \qquad w = 1,2,\dots,W \qquad (9)$$

$$C_{max} \geq C_{jT} \qquad j = 1,2,\dots,n \qquad (10)$$

$$Z_{iwt}, X_{ijt}, Y_{iljt} \in \{0,1\} \qquad w = 1,2,\dots,W, j = 1,2,\dots,n,\ t = 1,2,\dots,T, i = 1,2,\dots,S_t \qquad (11)$$

The objective function is considered to minimize the maximum completion time (makespan). Constraint set (2) ensures that each job must be assigned to only one machine at each stage. Constraint set (3) presents the completion time of job j in the first stage, and constraint set (4) is incorporated into the model to present the relation between completion times in two consecutive stages for job j. Constraint sets (5) and (6) preclude the interference between the processing of any two jobs on a machine at each stage. At most, one of these constraint sets is active for each pair of jobs. If job l is processed before job j on the same machine in stage i, constraint set (5) is activated to prevent and constraint set (6) will be redundant. On the contrary, the roles of these two constraint sets are changed. Constraint set (7) shows that the processing time of a job depends on the worker assigned to its machine. Constraint sets (8) and (9) define the assignment of the workers to each machine at each stage, and Constraint (10) defines C_{max} as the completion time of the last job. Finally, through constraint set (11), the binary variables are defined.

In the above mentioned mathematical model, the decision variables are multiplied together in the constraint sets (3)-(6), and these constraints are obviously nonlinear. Here, the nonlinear constraints are linearized in order to enhance the efficiency of our proposed mathematical model. For linearization, a new auxiliary binary variable is introduced as follows:

$$P''_{ijt} = P'_{ijt} X_{ijt} \qquad j = 1,2,\dots,n;\ t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (12)$$

In order to linearize the nonlinear term, the following equations are applied:

$$P''_{ijt} \leq P'_{ijt} \qquad j = 1,2,\dots,n;\ t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (13)$$

$$P''_{ijt} \leq M X_{ijt} \qquad j = 1,2,\dots,n;\ t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (14)$$

$$P_{ijt}^{"} \geq P_{ijt}' - M(1 - X_{ijt}) \qquad j = 1,2,\dots,n;\ t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (15)$$

$$P_{ijt}^{"} \geq 0 \qquad j = 1,2,\dots,n;\ t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (16)$$

Considering constraint sets (12)-(16), the linearized form of constraint sets (3)–(6) is as follows:

$$C_{j1} \geq \sum_{i=1}^{S_1} P_{ij1}^{"} \qquad j = 1,2,\dots,n \qquad (17)$$

$$C_{jt} \geq C_{j,t-1} + \sum_{i=1}^{S_t} P_{ijt}^{"} \qquad j = 1,2,\dots,n; t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (18)$$

$$C_{jt} + M\left(3 - X_{ijt} - X_{ilt} - Y_{iljt}\right) \geq C_{lt} + P_{ijt}^{"} \qquad j,l = 1,2,\dots,n; j \neq l; t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (19)$$

$$C_{lt} + M\left(2 - X_{ijt} - X_{ilt} + Y_{iljt}\right) \geq C_{jt} + P_{ilt}^{"} \qquad j,l = 1,2,\dots,n; j \neq l; t = 1,2,\dots,T;\ i = 1,2,\dots,S_t \qquad (20)$$

3-3. Metaheuristic algorithms

The MILP model, presented in this paper, is capable of solving the small-sized considered problem in a reasonable amount of time. Since this problem is NP-hard, it is inevitable to use metaheuristic approaches to solve the problem under investigation, approximately. Thus, two metaheuristic algorithms based on the PSO are developed in this research.

3-3-1. PSO algorithm

Particle Swarm Optimization (PSO) is a population-based metaheuristic algorithm, which was originally proposed by Kennedy James and Russ Eberhart in 1995. It was firstly developed to optimize the non-linear functions in the continuous search space. The PSO algorithm includes particles that try to improve them in an n-dimensional search space according to simple mathematical formulae. Each particle moves around in the solution space for the optimal solution by updating its velocity and position based on two parts: cognition part and social part. The following formulae are used in every iteration for updating the velocity and position of a particle [30]:

$$vel_i(k+1) = W \times vel_i(k) + C_1 \times r_1 \times \left(pbest_i - X_i(k)\right) + C_2 \times r_2 \times (gbest - X_i(k)) \qquad (21)$$

$$X_i(k+1) = X_i + vel_i(k+1) \qquad (22)$$

where w is called inertial weight and shows the influence of the previous velocity of the particle on its velocity in the next iteration. $vel_i(k)$ represents the velocity of the i th particle in iteration k, and X_i (k) shows the position of particle i in the kth iteration. $pbest_i$ and $gbest$ are the best-known position vector of particle i and the best location vector in the whole population for the entire particles, respectively. Parameters C_1 and C_2 are acceleration coefficients with constant values and determine the influence of $pbest$ and $gbest$ values on the velocity, respectively. Two random numbers, r_1 and r_2, are incorporated into the structure of the PSO algorithm to add uncertainty.

According to Poli et al. [31], incorporating multiplier (χ) into Eq. (21) can speed up the convergence process and enhance the performance of the PSO algorithm. The proper value of χ is calculated as Eq. (23):

$$\chi = \frac{2}{C - 2 + \sqrt{C^2 - 4C}}, (C = c_1 + c_2 > 4) \qquad (23)$$

Based on Eq. (23), Eq. (24) is applied to update the velocity of each particle in all iterations:

$$vel_i(k+1) = \chi \times [w \times vel_i(K) + C_1 \times r_1 \times \left(pbest_i - X_i(k)\right) + C_2 \times r_2 \times \left(gbest - X_i(k)\right)] \qquad (24)$$

Similar to Tadayon and Salmasi [25], Eq. (25) is used to determine the value of *w* in each iteration.

If *w* is assigned a high value at the beginning of the procedure and *w* is gradually reduced to a lower value, better performance of the search procedure can be ensured.

$$w = w_{max} - \frac{(w_{max} - w_{min}) * iter}{max_{iter}} \quad (25)$$

where w_{max} and w_{min} are the upper and lower bounds for *w*, respectively, and max_{iter} is the total number of iterations that accomplished in the PSO algorithm.

3-3-2. Particle encoding approach

As mentioned above, the original PSO algorithm is proposed to solve the continuous problem. On the other hand, job scheduling and worker assignment are discrete problems in the considered research problem. Thus, it is necessary to propose a suitable procedure to transform a vector in continuous space to the one in discrete space. In this paper, the smallest value (SV) technique is applied for this conversion. According to this technique, the sequence can be assigned by a non-decreasing order of the values in the original vector. In order to generate a sequence based on the original vector, entire values in the vector are sorted in non-decreasing order. Then, the position of the values in the new order, in comparison to the original vector, forms the job sequence. A simple example presented in Section 3.2.5 shows the construction of a discrete-values particle by the SV rule.

3-3-3. Hybrid SA and PSO algorithm (SPSO algorithm)

Due to the trapping of the PSO in local optima in some cases, PSO has been modified to tackle this difficulty by incorporating the Boltzmann-type operator into the standard PSO. The proposed procedure can prevent the convergence of the PSO. The early convergence of the fast clustering of the particles occurs near the optimal solution, whereas the optimal solution might be a local optimum. The Boltzmann-type operator in the SA algorithm can prepare better variety in searching the solution space.

The idea is that if the $gbest$ ($pbest$) of the current iteration (particle) has better performance, the new particle can be accepted; however, if the $gbest$ ($pbest$) is inferior, we may still accept it with a positive probability. In doing so, a random number, *P*, between 0 and 1 is generated and, then, the $gbest$ ($pbest$) will be accepted if:

$$exp\left(\frac{C(x_t) - C(x_{t-1})}{T}\right) \geq P \quad (26)$$

In this inequality, $C(x_t)$ is the objective value of $gbest$ ($pbest$) in the current iteration (*t*), $C(x_{t-1})$ is the objective value of its previous solution, and *T* denotes the temperature (analogous to physical annealing) at which the current solution is evaluated. *T* is a function of two input parameters: initial temperature and cooling rate. The pseudo-code of the hybrid approach is presented in Fig. 2 as follows:

Initialization

Population size ($popsize$)/number of iteration ($maxiter$)/constant values (C_1, C_2)/maximum inertial weight (w_{max})/minimum inertial weigh (w_{min})/initial temperature ($initemp$)/cooling rate (α)

Generate initial population for job scheduling and worker assignment according to SV technique

Set $k = 1$

While $k \leq maxiter$ Do

Calculate the objective value for particle i ($Cmax_i^k$)

Evaluate the particles to get $pbest_i^k$

if $Cmax_i^k \leq pbest_i^{k-1}$ then $pbest_i^k = Cmax_i^k$

else if $r \leq e^{\frac{(Cmax_i^k - pbest_i^{k-1})}{T}}$ then $pbest_i^k = Cmax_i^k$

else $pbest_i^k = pbest_i^{k-1}$

Evaluate the particles to get $pbest^k$

if $\min_i\{Cmax_i^k\} \leq gbest^{k-1}$ then $gbest^k = \min_i\{Cmax_i^k\}$

else if $r \leq e^{\frac{(\min_i\{Cmax_i^k\} - gbest^{k-1})}{T}}$ then $gbest^k = \min_i\{Cmax_i^k\}$

else $gbest^k = gbest^{k-1}$

Reduce temperature $T = T \times \alpha$

$k = k + 1$

End while

Fig. 1. Pseudo-code of the hybrid approach

The termination criteria in both PSO and SPSO algorithms are based on the maximum iteration.

3-3-4. Calculating the objective function

In order to calculate the objective function value for a given sequence, a decision should be made about the assignment of the workers to each machine, firstly. For this purpose, according to the initial sequence of the workers, generated by SV technique, entire machines in all stages are arranged in a $1 \times \left|\sum_{t=1}^{T} s_t\right|$ vector (machine vector), and each worker is assigned to the corresponding machine at machine vector. The second decision is to assign the jobs to each machine at each stage. In this research, for a given sequence, the job in the sequence is assigned to all of the available and unavailable machines at each stage and a machine that has the earliest completion time is selected. Because of the unrelated parallel machines at each stage, an unavailable, yet more efficient, machine may produce an earlier completion time for a given job; in other words, this rule may prefer an unavailable machine with short processing time to an available machine with long processing time.

3-3-5. A simple example

In order to determine how the worker assignment and job scheduling are generated based on the proposed metaheuristic approaches, a simple example is considered. Suppose that there is a scheduling problem with 5 jobs, 6 workers, and 2 machines in 3 stages. The workers have different processing times for the same jobs, which have produced randomly at the interval [2,10]. Table 2 shows the processing time of each job on each machine at each stage by each worker.

Tab. 2. Data of example

			Job 1	Job 2	Job 3	Job 4	Job 5
	Stage 1	Machine 1	6	7	3	6	4
		Machine 2	6	8	8	8	8
Worker 1	Stage 2	Machine 1	2	8	9	9	7
		Machine 2	10	9	3	7	7
	Stage 3	Machine 1	5	2	9	7	2
		Machine 2	5	10	8	7	10
	Stage 1	Machine 1	9	7	6	6	3
		Machine 2	2	6	6	3	8
Worker 2	Stage 2	Machine 1	7	7	10	4	2
		Machine 2	10	5	2	5	10
	Stage 3	Machine 1	4	3	4	9	10
		Machine 2	8	6	4	7	5
	Stage 1	Machine 1	8	3	10	10	2
		Machine 2	5	10	2	4	5
Worker 3	Stage 2	Machine 1	10	2	7	7	8
		Machine 2	5	5	7	7	8
	Stage 3	Machine 1	10	6	9	9	4
		Machine 2	8	5	8	3	6
	Stage 1	Machine 1	7	8	9	9	9
		Machine 2	6	4	7	3	7
Worker 4	Stage 2	Machine 1	10	7	2	2	2
		Machine 2	4	3	3	3	3
	Stage 3	Machine 1	2	7	5	6	3
		Machine 2	8	4	9	4	4
	Stage 1	Machine 1	10	6	3	6	3
		Machine 2	10	2	4	2	10
Worker 5	Stage 2	Machine 1	2	6	10	4	7
		Machine 2	6	4	3	5	4
	Stage 3	Machine 1	5	10	4	8	9
		Machine 2	10	7	6	8	10
	Stage 1	Machine 1	4	4	9	6	6
		Machine 2	8	3	4	7	8
Worker 6	Stage 2	Machine 1	8	6	10	10	9
		Machine 2	4	2	2	4	10
	Stage 3	Machine 1	3	3	9	8	10
		Machine 2	5	5	10	5	7

At first, it is supposed that the proposed metaheuristic approach generates a vector for the job sequence as J= (0.5,0.2,0.3,0.8,0.9) in an iteration. The equivalent sequence vector generated by the SV technique should be: Seq=(3,1,2,4,5). The smallest value in vector J is 0.2, which is located at the second position in J vector; therefore, the second position in vector Seq should be 1. The second smallest value in vector J is 0.3, which is located at the third position in vector J; therefore, the third position in vector Seq should be 2. Other values in vector Seq are calculated in a similar manner. As the same way, the sequence of the worker can be calculated. Suppose that, in an iteration, the random vector is as W=(0.7, 0.3, 0.6, 0.8, 0.1, 0,4). According to the smallest value, the sequence of workers is W= (5,2,4,6,1,3). Therefore, each worker is assigned to each machine from the first machine in the first stage to the last machine in the last stage.

As mentioned above, in order to assign jobs to each machine in each stage, a job in the sequence is assigned to all the available and unavailable machines in each stage and a machine that has the earliest completion time is selected. With respect to the workers' sequence, the processing time of the jobs in each stage on each machine is extracted from Table 2. As a result, the Gant chart of the generated solution is shown in Figure 2.

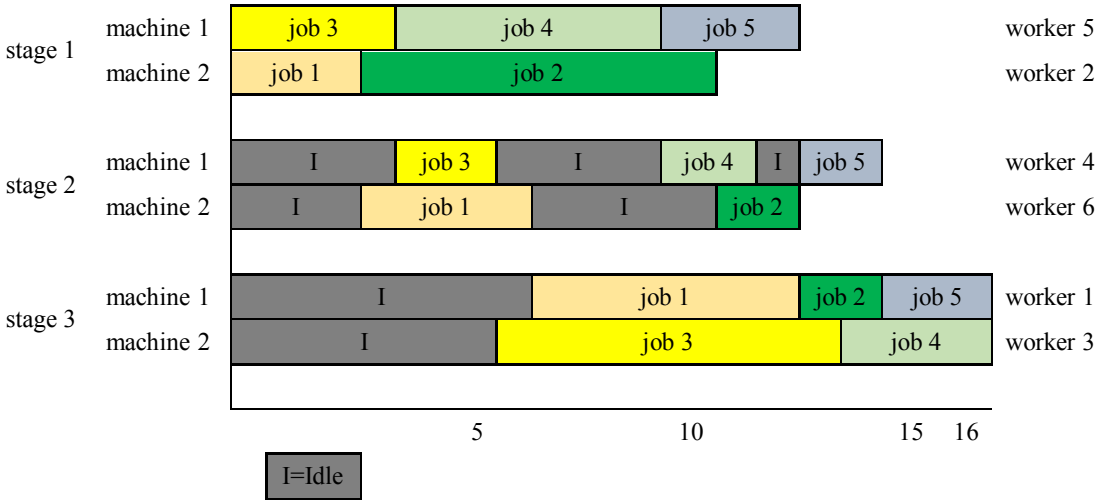


Fig. 2. Worker assignment and job scheduling schedule

4. Discussion

The computational study aims to evaluate the performances of the MILP model and the proposed metaheuristic approaches in minimizing the makespan for the FFS scheduling problem with worker assignment based on some test problems. The metaheuristic algorithms have been coded in MATLAB 12.1 and run on 2.1 GHz Laptop with 2 GB of RAM. The MILP model is also coded in Lingo software and solved by CPLEX solver.

At first, the input parameters for the metaheuristic algorithms are set and, then, the performances of these algorithms are compared with that of the optimal solution in small-size problems. At last, an attempt has been made to consider the superiority of metaheuristic algorithms based on large-size test problems.

4-1. Selecting the best input parameters

Selecting the best values for the input parameters in the structure of the metaheuristic algorithms can significantly enhance the performance of these algorithms. This section describes an empirical testing approach to achieving suitable input parameters for the PSO and SPSO algorithms. As mentioned above, the PSO and SPSO algorithms have some input parameters. The initial values of these input parameters are selected based on the experience of other research, as discussed in the literature. In order to find the suitable values for these parameters, 5 test problems are generated, i.e., two small and three large-size problems, and each test problem is solved by each combination of the input parameters ($2^2 \times 3^5$ combinations). Thus, we solved $5 \times 2^2 \times 3^5 = 4860$ to derive the best input parameters. Table 3 shows the initial and best values of these input parameters.

Tab. 3. Initial and best input parameters for the PSO and SPSO algorithms

Parameters	Initial values	Best value
C_1,C_2	2.1, 2.2, 2.3, 2.4	2.4
W_{max}	0.9, 0.8, 0.7	0.9
W_{min}	0.4, 0.3	0.3
Popsize	20, 30, 50	30
Maxiter	500, 800, 1000	500
T	25, 50, 100	50
α	0.9,0.99	0.99

4-2. Comparison of the proposed metaheuristic approaches and the optimal solution

In this section, we focus on the comparison of the optimal solutions obtained from the MILP model and the proposed PSO and SPSO algorithms for the small-size problems. For this purpose, 15 examples of small size are generated. Three characteristics, i.e., the number of jobs, number of stages, and number of the unrelated parallel machines in each stage, are used to typify each test problem. The number of workers in each

stage is equal to that of machines in the same stage, because each worker must be assigned to one machine. Thus, to show each test problem, a three-section notation, $J_a S_b M_c$, is applied. The first section, J_a, expresses the number of jobs, S_b shows the number of stages, and the last section demonstrates the number of machines in each stage.

Based on the small-sized test problem, when the number of jobs increases up to 6, the MILP requires extensive CPU time, and it is rapidly increasing on the problem scale. Thus, the results include only the test problems, which can obtain the optimal solution at a time limit of 3600s. The comparison of the proposed metaheuristic approaches and the optimal solution is presented in Table 4. The results in Table 4 include the optimal solution, the makespan, CPU time, and optimal Gap for each proposed metaheuristic. Note that the MILP is modeled in Lingo software and is solved by the Cplex solver.

The optimal gap can be calculated as follows:

$$Optimal\ Gap = \frac{Metaheuristic_{Makespan} - Optimal_{Makespn}}{Optimal_{Makespn}} \quad (27)$$

Tab. 4. The comparison of Lingo, PSO, and SPSO algorithms for small-sized problems

Problem	Problem structure	Optimal solution		PSO			SPSO		
		C_{max}	CPU Time	C_{max}	CPU Time	Optimal Gap	C_{max}	CPU Time	Optimal Gap
1	$J_2S_2M_2$	57	1	57	3.8	0.0%	57	4.2	0.0%
2	$J_2S_3M_2$	79	2	80	6.5	1.3%	79	6.8	0.0%
3	$J_2S_2M_3$	48	1	48	6.0	0.0%	48	6.0	0.0%
4	$J_3S_2M_2$	73	3	75	5.9	2.7%	74	6.4	1.4%
5	$J_3S_3M_2$	101	10	105	6.2	4.0%	103	7.0	2.0%
6	$J_3S_2M_3$	49	3	50	7.0	2.0%	50	7.0	2.0%
7	$J_4S_2M_2$	78	9	79	5.5	1.3%	78	5.6	0.0%
8	$J_4S_3M_2$	109	25	112	6.0	2.8%	110	6.0	0.9%
9	$J_4S_2M_3$	68	8	68	7.8	0.0%	68	8.3	0.0%
10	$J_5S_2M_2$	95	70	96	8.2	1.1%	95	8.8	0.0%
11	$J_5S_3M_2$	111	91	115	8.6	3.6%	113	9.1	1.8%
12	$J_5S_2M_3$	81	57	82	7.3	1.2%	81	7.5	0.0%
13	$J_6S_2M_2$	122	1087	125	9.2	2.5%	122	9.9	1.6%
14	$J_6S_3M_2$	204	2011	210	9.5	2.9%	209	9.8	2.5%
15	$J_6S_2M_3$	92	316	95	8.9	3.3%	94	10.1	2.2%

According to Table 4, the PSO algorithm can achieve the optimal solution in 20% of test problems. In addition, the average difference between the PSO and optimal solution (optimal gap) is equal to 1.9%. However, the SPSO algorithm can generate the optimal solution in 53% of the test problems, and the average optimal gap is equal to 0.85%. Moreover, the average CPU times for the PSO and SPSO are 7.1 and 7.5 seconds, respectively. The results show that both of metaheuristic approaches can obtain the optimal and near-optimal solutions in a reasonable amount of time, yet SPSO performs more efficiently.

4-3. Evaluation of PSO and SPSO algorithms in the large-size problems

In order to confirm which algorithm has the best performance, this section is devoted to evaluating and comparing the proposed metaheuristic algorithms based on some test problems on medium and large scales. For this purpose, we apply some test problems of different sizes and compare these algorithms based on Makepsn and CPU time. The generation of the test problems is described as follows: The numbers of jobs are extended to 10, 20, 30, 40, and 50. In addition, the number of stages includes 5, 7, and 9 and the number of machines in each stage belongs to the interval of [2,5]. Processing times are generated randomly in the range of [5,100].

The experimental results include the makespan, CPU times, and PRM, as shown in Table 5. The PRM, the percentage of the reduction in the makespan for the SPSO compared to PSO, is calculated as follows:

$$PRM = \frac{PSO_{Makespan} - SPSO_{Makespn}}{PSO_{Makespn}} \times 100 \qquad (28)$$

Tab. 5. The computational results

			PSO		SPSO						PSO		SPSO		
instance	Job	stage	Makespan	CPU Time	Makespan	CPU Time	PRM	instance	Job	stage	Makespan	CPU Time	Makespan	CPU Time	PRM
1			514	22.3	486	19.6	5.4%	51			3336	64.7	3057	60.8	8.4%
2			462	19.8	441	19.5	4.5%	52		7	3255	64.5	3072	63.3	5.6%
3			438	20.4	429	18.9	2.1%	53			3320	65.2	3104	62.9	6.5%
4		5	466	20.9	448	20.2	3.9%	54			3534	68.3	3232	63.0	8.5%
5			581	19.8	553	18.9	4.8%	55	30		4243	71.7	3855	67.0	9.1%
6			519	19.6	500	18.9	3.7%	56			4203	71.2	3852	66.0	8.4%
7			524	19.9	500	19.3	4.6%	57		9	3416	75.2	3110	69.2	9.0%
8			898	27.7	854	26.2	4.9%	58			3760	72.8	3481	69.9	7.4%
9			726	23.0	711	22.9	2.1%	59			3489	74.9	3265	70.9	6.4%
10	10		762	22.7	722	22.4	5.2%	60			3264	75.7	3067	71.3	6.0%
11		7	821	22.9	778	22.1	5.2%	61			4208	88.1	3903	88.3	7.2%
12			943	22.2	919	21.9	2.5%	62			3761	79.3	3545	75.7	5.7%
13			721	23.9	706	23.4	2.1%	63			4409	82.2	4007	77.5	9.1%
14			618	24.7	605	23.8	2.1%	64		5	4060	79.1	3757	76.5	7.5%
15			1117	28.4	1064	27.9	4.7%	65			4046	78.8	3679	74.1	9.1%
16			1274	27.9	1206	27.9	5.3%	66			3838	81.6	3472	77.7	9.5%
17		9	1196	28.7	1129	27.5	5.6%	67			3825	81.8	3490	77.2	8.8%
18			1177	27.7	1108	27.6	5.9%	68			4261	89.6	3928	84.6	7.8%
19			1154	28.4	1104	27.6	4.3%	69			4377	89.4	3995	84.6	8.7%
20			1180	29.1	1110	27.6	5.9%	70			4543	88.6	4180	83.7	8.0%
21			1676	37.0	1596	37.1	4.8%	71	4	7	4207	88.2	3945	83.7	6.2%
22			1601	38.2	1507	37.2	5.9%	72			4296	88.9	3980	83.6	7.4%
23			1662	37.8	1570	37.4	5.5%	73			4705	89.5	4391	83.7	6.7%
24		5	1589	38.1	1540	36.6	3.1%	74			4418	90.7	4166	87.1	5.7%
25			1889	35.5	1790	34.0	5.2%	75			4738	92.7	4358	89.8	8.0%
26			1547	37.6	1508	36.6	2.5%	76			5056	92.0	4599	88.0	9.0%
27			1597	37.7	1511	35.6	5.4%	77		9	3826	92.8	3569	90.7	6.7%
28			1592	42.0	1531	39.5	3.8%	78			4576	97.3	4186	91.2	8.5%
29			1847	41.5	1800	39.7	2.5%	79			4333	97.9	3966	91.2	8.5%
30	20		1847	41.4	1720	40.2	6.9%	80			4903	96.8	4608	90.5	6.0%
31		7	1751	41.7	1618	40.3	7.6%	81			4735	103.7	4402	93.2	7.0%
32			1795	42.1	1674	40.7	6.7%	82			5028	97.1	4653	92.8	7.5%
33			1805	42.8	1653	41.7	8.4%	83			4629	98.1	4225	93.4	8.7%
34			1985	44.4	1849	41.7	6.9%	84		5	4966	99.9	4560	93.4	8.2%
35			2152	46.4	2066	43.6	4.0%	85			5181	103.6	4780	96.3	7.7%
36			1936	47.0	1822	43.6	5.9%	86			4648	101.3	4210	95.3	9.4%
37		9	1825	50.1	1733	44.6	5.0%	87			4669	100.9	4236	95.0	9.3%
38			2033	49.7	1926	47.0	5.3%	88			5053	102.4	4625	112.6	8.5%
39			2008	49.5	1855	46.6	7.6%	89			6035	102.9	5545	113.2	8.1%
40			1974	51.3	1859	46.9	5.8%	90	50		5975	101.1	5516	111.2	7.7%
41			2643	52.1	2446	50.4	7.5%	91		7	5933	102.5	5445	112.8	8.2%
42			2292	51.9	2213	50.5	3.4%	92			5727	101.8	5338	112.0	6.8%
43			2469	51.3	2256	50.5	8.6%	93			6787	104.0	6197	114.4	8.7%
44		5	2393	51.4	2197	50.7	8.2%	94			5916	106.1	5394	116.7	8.8%
45	30		2666	48.3	2543	45.9	4.6%	95			7175	106.8	6693	117.5	6.7%
46			2527	50.7	2330	47.6	7.8%	96			5828	109.6	5329	120.6	8.6%
47			2361	54.1	2179	50.6	7.7%	97		9	6649	111.7	6046	122.9	9.1%
48			2680	62.3	2471	61.1	7.8%	98			6640	109.5	6043	120.5	9.0%
49		7	3599	65.5	3345	61.1	7.1%	99			7364	111.3	6861	122.4	6.8%
50			3136	64.4	2902	60.2	7.5%	100			7533	108.5	6890	119.4	8.5%

According to Table 5, it is observed that the proposed SPSO algorithm gives smaller PRM in the entire test problems. Thus, we can conclude that the application of the Boltzmann operator in the structure of the PSO algorithm can reduce the makespan by PRM = 6.6% on average. Moreover, there is not any significant difference between CPU times of the proposed algorithms to solve the research problems. On the total average PRM ($\overline{PRM}$), an improvement of 4.2% with respect to Job=10, 5.4%, Job=20, 7.3%, Job=30, 7.9%, and Job=40, and an improvement of 8.2% with respect to Job= 50 has been achieved for the various test problems considered in this study. In addition, the $\overline{PRM}$ for 5, 7, and 9 stages is equal to 6.5%, 6.4%, and 6.9%, respectively. Therefore, the SPSO can be stated as a key algorithm, compared to PSO, to solve the simultaneous worker assignment and scheduling problem in the FFS with unrelated parallel machines.

Moreover, at a significance level of 5%, a one-sample t-test is applied to investigate whether or not the average makespan obtained by the SPSO is smaller than that by the PSO ($\overline{PRM} = 0\ vs.\ \overline{PRM} \neq 0$). With a P-value of 0.000, the experimental results show that the SPSO algorithm outperforms the PSO algorithm for large-size problems.

In order to evaluate the effect of various experimental factors used in test problems on the solution quality (PRM), the effects of the problem category on $\overline{PRM}$ are illustrated. Two characteristics are used to represent a problem category such as the number of jobs and stages.

4-4. Analysis of parameters of test problems

This section is devoted to analyzing the effect of parameters of test problem on the proposed metaheuristic algorithms. For this purpose, two main parameters, number of jobs and number of stages, are selected.

Analysis of number of jobs: in order to investigate the effect of the number of jobs on the proposed algorithms, the interaction between the algorithms and number of jobs is illustrated in Figure 3. As is observed, by increasing the number of jobs, the average PRM is also increased.

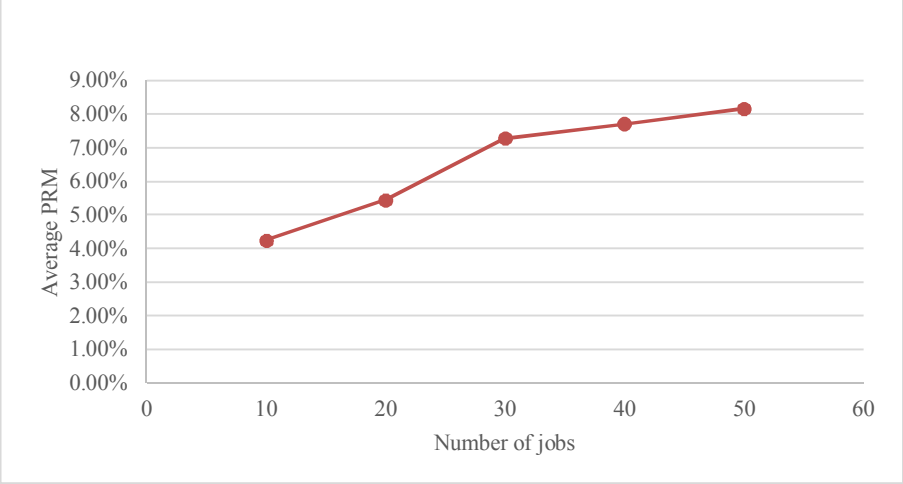


Fig. 3. The interaction between the number of jobs and average PRM

Analysis of number of stages: another plot for interaction between the algorithms and number of stages is shown in Figure 4. Regarding Figure 4, the average PRM is also increased by increasing the number of stages.

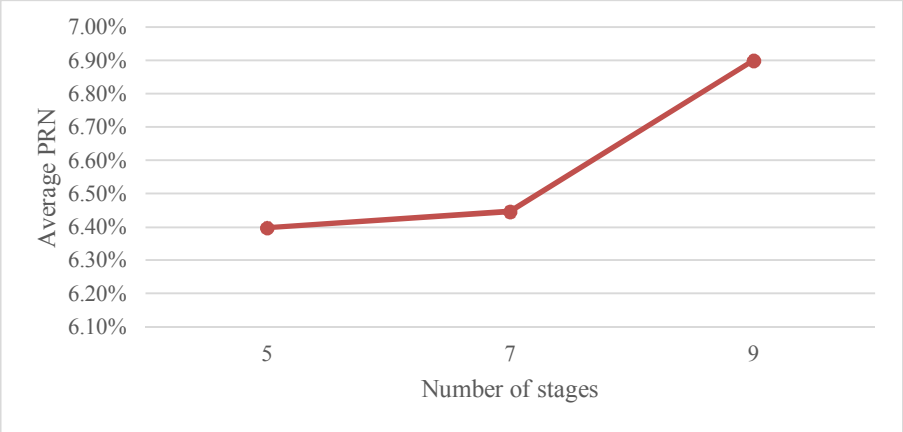
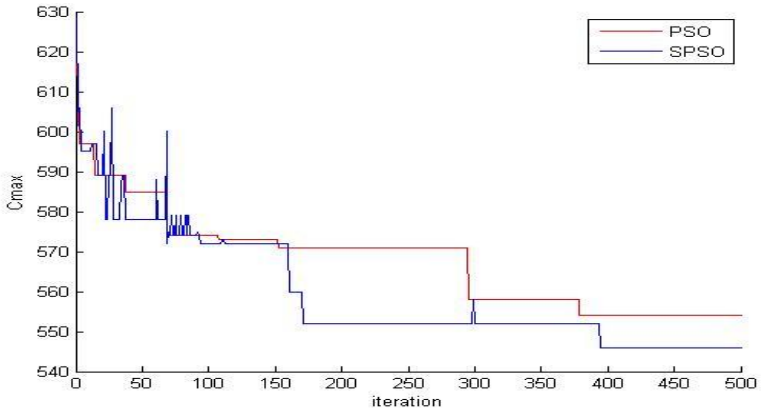


Fig. 4. Effect of experimental factors on solution quality

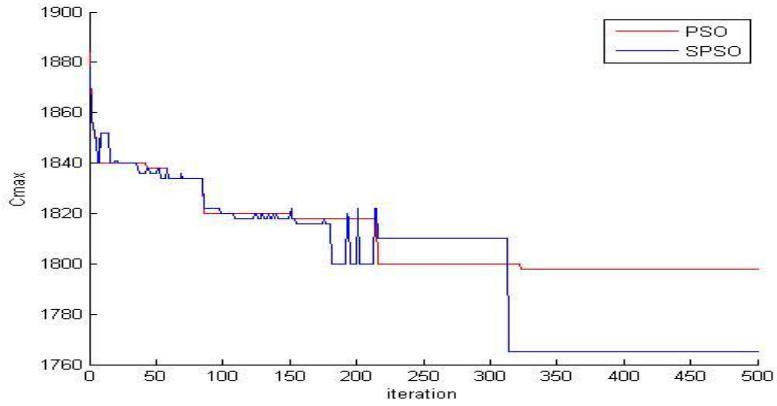
As can be seen in Figures 3 and 4, the superiority of the SPSO algorithm gets more significant with an increase in the number of jobs and number of stages in most categories.

4-5. Convergence trend of the proposed metaheuristic algorithms

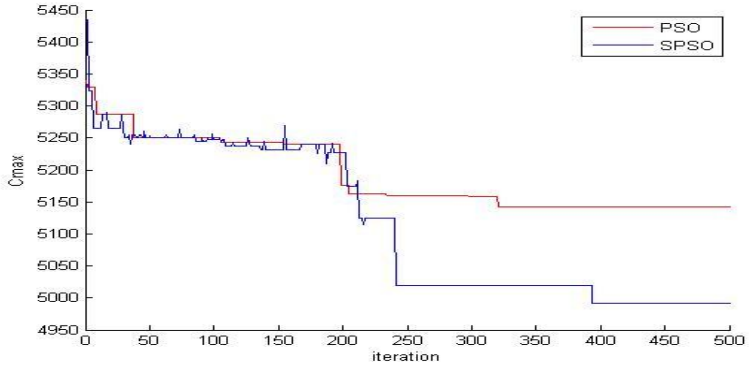
Figure 4 illustrates the convergence trend of the solutions for three test problems with 10, 20, and 50 jobs. From the figure, we can observe that the PSO algorithm can quickly obtain better solutions at the beginning of the search procedure; however, this solution is local optima with high probability. On the other hand, the SPSO searches the inferior solutions at the beginning of the procedure to avoid the local optima, and it can lead to near-optimal solutions at the end of the procedure.



Jobs=10



Jobs=20



Jobs=50

Fig. 5. Convergence plots for proposed metaheuristic approaches

Based on these results, it can conclude that the proposed SPSO is more effective and robust than the PSO algorithm to identify better solutions for simultaneous workers' assignment and scheduling problem in the FFS with unrelated parallel machines.

5. Conclusions and Future Research

This study considered a simultaneous worker assignment and flexible flow shop scheduling problem in which we found the best worker assignment to each machine in each stage firstly and, then, solved the corresponding scheduling problem. In this problem, the processing time of a job on a machine at any stage is dependent on the assigned worker. This study presented an MILP model with makespan objective function to determine the optimal worker assignment and scheduling problem. Because of the NP-hardness of the problem, two metaheuristic algorithms, PSO and SPSO, were proposed to solve the problem, heuristically. The results of the test problem revealed that the PSO algorithm armed with Boltzmann operator (SPSO) was more effective and efficient for both small- and large-sized problems. Because of the novelty of the research problem in the context of the flexible flow shop problem, there is no benchmark problem to evaluate the efficiency of the proposed metaheuristics. As a result, some test problems for this purpose were developed.

The research problem has some applications in the real-world industry, encountering the worker assignment in the production line. By applying the worker assignment in the other scheduling problems such as job shops and open shops, other metaheuristic approaches were proposed to solve the research problem, and considering other objective functions such as T_{max} can be recommended as future research areas. Considering other assumptions in the research problems, such worker-based processing times, sequence-dependent processing time, and eligible worker constraints can be other clues of the future researches.

References

[1] Arthanari, T. S., Ramamurthy, K. G. "An extension of two machines sequencing problem" Opsearch, Vol. 8, No. 1, (1971), pp. 10-22.

[2] Gaffari, E., Sahraeian, R. "A Two-Stage Hybrid Flowshop Scheduling Problem with Serial Batching". International Journal of Industrial Engineering & Production Research. Vol. 28, No. 1, (2014), pp. 55-63.

[3] Asadi-Gangraj, E. "A Heuristic approach to solve hybrid flow-shop scheduling problem with unrelated parallel machines". International Journal of Industrial Engineering & Production Research. Vol. 28, No. 1, (2017), pp. 61-74.

[4] Nahavandi, N., Asadi-Gangraj, E. "A new lower bound for flexible flow shop scheduling problem with unrelated parallel machines". International Journal of Industrial Engineering & Production

Research. Vol. 25, No. 1, (2014), pp. 65-70.

[5] Asadi-Gangraj, E. "Lagrangian relaxation approach to minimize makespan for hybrid flow shop scheduling problem with unrelated parallel machines", Scientia Iranica, Vol. 25, No. 6, (2018), pp. 3765-3775.

[6] Behnamian, J. "Scheduling and worker assignment problems on hybrid flowshop with cost-related objective function". International Journal of Advanced Manufacturing Technology, Vol. 74, No. 1, (2014), pp. 267–283.

[7] Alizadeh, M., Mahdavi, I., Mahdavi-Amiri, N., Shiripour, S. "A capacitated location-allocation problem with stochastic demands using sub-sources: An empirical study". Applied Soft Computing, Vol. 34, (2015), pp. 551-571.

[8] Benavides, A., Ritt, J. M., Miralles, C. "Flow shop scheduling with heterogeneous workers". European Journal of Operational Research, Vol. 237, No. 2, (2014), pp. 713-720.

[9] Celano, G., Costa, A., Fichera, S. "Makespan minimization in a flowshop sequence dependent group scheduling and worker assignment problem". Proceedings of the 41st International Conference on Computers and Industrial Engineering, Vol. 25, (2014), pp. 254-259.

[10] Tyagi, N., Seidgar, H., Abedi, M., Chandramouli, A.B. "Learning and Forgetting Effects of Flexible Flow Shop Scheduling" International Journal of Innovation and Applied Studies, Vol. 7, No. 3, (2014), pp. 857-867.

[11] Hu, P-C. Minimizing total tardiness for the worker assignment scheduling problem in identical parallel-machine models, International Journal of Advanced Manufacturing Technology, Vol. 23, (2004), pp. 383-388.

[12] Hu, P-C. "Minimizing total flow time for the worker assignment scheduling problem in the identical parallel-machine models". International Journal of Advanced Manufacturing Technology, Vol. 25, (2005), pp. 1046-1052.

[13] Hu, P-C. "Further study of minimizing total tardiness for the worker assignment scheduling problem in the identical parallel machine models". International Journal of Advanced Manufacturing Technology, Vol. 29, (2006), pp. 165-169.

[14] Hu, P-C. "Further study of minimizing total flowtime for the worker assignment scheduling problem in the identical parallel machine models". International Journal of Advanced Manufacturing Technology, Vol. 29, (2006), pp. 753-757.

[15] Chaudhry, I. A., Drake, P .R. "Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms". International Journal of Advanced Manufacturing Technology, Vol. 42, (2009), pp. 581-594.

[16] Chaudhry, I. A. "Minimizing flow time for the worker assignment problem in identical parallel machine models using GA". International Journal of Advanced Manufacturing Technology, Vol. 48, (2010), pp. 747-760.

[17] Carniel, G. C., Benavides, A. J., Ritt, M. "Models for the inclusion of workers with disabilities in flow shop scheduling problems". Simpósio Brasileiro de Pesquisa Operacional. (2013), pp. 3320-3329.

[18] Aftab, M. T., Muhammad, U., Riaz, A. "Jobs scheduling and Worker Assignment Problem to Minimize Makespan using Ant Colony Optimization Metaheuristic". International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, Vol. 72, (2012), pp. 2823-2826.

[19] Zabihzadeh, S., Rezaeian, J. "Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time". Applied Soft Computing, Vol. 40, (2016), pp. 319-330.

[20] Rabiee, M., Jolai, F., Asefi, H., Fattahi, P., Lim, S. "A biogeography-based optimisation algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness". International Journal of Computer Integrated Manufacturing, Vol. 29, No. 9, (2016), pp. 1007-1024.

[21] Mousavi, S. M., Zandieh, M. "An Efficient Hybrid Algorithm for a Bi-objectives Hybrid Flow Shop Scheduling". Intelligent automation & Soft Computing, DOI: 10.1080/10798587.2016.1261956, (2016).

[22] De Siqueira, E. C., Souza, M. J. F., De Souza, S.R. "A Multi-objective Variable Neighborhood Search algorithm for solving the Hybrid Flow Shop Problem". Electronic Notes in Discrete Mathematics, Vol. 66, (2018), pp. 87-94.

[23] Asadi Gangraj, E., Nahavandi, N. "A metaheuristic approach for batch sizing and scheduling problem in flexible flow shop with unrelated parallel machines". International Journal of Computer Applications, Vol. 97, No. 6, (2014), pp. 31-36.

[24] Rajaee Abyaneh, F., Gholami, S. "A comparison of algorithms for minimizing the sum of earliness and tardiness in hybrid flow-shop scheduling problem with unrelated parallel machines and sequence-dependent setup times". Journal of Industrial and System Engineering, Vol. 8, No. 2, (2015), pp. 67-85.

[25] Tadayon, B., Salmasi, N. "A two-criteria objective function flexible flowshop scheduling problem with machine eligibility constraint". International Journal of Advanced Manufacturing Technology, Vol. 64, (2013), pp. 1001-1015.

[26] Ou, W., Zou, F., Gao, Z. "Flexible Flow-shop Scheduling Approach Based on Hybrid Particle Swarm Optimization". Control and Decision Conference, (2008), Chinese.

[27] Tang, D., Dai, M., Salado, M. A., Giret, A. "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization". Computers and Industrial Engineering, Vol. 81, (2015), pp. 82-95.

[28] Ranjan Singh, M., Mahapatra, S. S. "A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks". International Journal of Advanced Manufacturing Technology, Vol. 62, No. 1, (2012), pp. 267-277.

[29] Li, J. Q., Pan, Q. K., Mao, K. "Hybrid Particle Swarm Optimization for Hybrid Flowshop Scheduling Problem with Maintenance Activities". Doi:10.1155/2014/596850, (2014).

[30] Pongchairerks, P. "A self-tuning PSO for job-shop scheduling problems". European Journal of Operational Research, Vol. 19, No. 1, (2014), pp. 96 - 113.

[31] Poli, R., Kennedy, J., Blackwell, T. "Particle swarm optimization an overview". International Journal of Computer Integrated Manufacturing, Vol. 29, No. 9, (2007), pp. 1007-1024.

Follow This Article at The Following Site:

Asadi-Gangraj E, Bozorgnezhad F, Paydar M M. A simultaneous worker assignment and scheduling problem to minimize makespan in flexible flow shop via metaheuristic approaches. IJIEPR. 2019; 30 (2) :207-223
URL: http://ijiepr.iust.ac.ir/article-1-860-en.html