Firefly Algorithm for Flow Shop Optimization

M.K. Marichelvam, T. Prabaharan and M. Geetha

Abstract In this chapter, a recently developed bio-inspired meta-heuristic algorithm namely firefly algorithm (FA) is addressed to solve the flow shop scheduling problems with sequence dependent setup times which have been proved to be strongly NP-hard type of combinatorial optimization problems. Four different performance measures namely minimization of makespan, mean flow time, mean tardiness and number of tardy jobs are considered. Extensive computational experiments were carried out to compare the performance of the proposed FA on different random problem instances. The results indicate that the proposed FA is more effective than many other algorithms reported earlier in the literature.

Keywords Flow shop · Scheduling · NP-hard · Firefly algorithm (FA) · Makespan · Flow time · Tardiness · Tardy jobs

1 Introduction

Scheduling is defined as a process of allocating resources over time to perform a collection of tasks [3]. It is a decision-making process and plays a vital role for the development industries. Scheduling problems are non-deterministic polynomial time hard (NP-hard) type combinatorial optimization problems. Hence it is difficult to solve the problems. Researchers addressed different type of scheduling problems [44]. Among them flow shop scheduling problems have attracted the researchers for the past several decades. Many industries such as metal, plastic, chemical and food

M.K. Marichelvam (✉) · T. Prabaharan
Department of Mechanical Engineering, Mepco Schlenk Engineering College, Sivakasi 626005, Tamilnadu, India
e-mail: mkmarichelvamme@gmail.com

M. Geetha
Department of Mathematics, Kamaraj College of Engineering and Technology, Virudhunagar 626001, Tamilnadu, India

X.-S. Yang (ed.), *Recent Advances in Swarm Intelligence and Evolutionary Computation*, Studies in Computational Intelligence 585, DOI 10.1007/978-3-319-13826-8_12

industries resemble the flow shop environment. Most of the research papers addressed the flow shop scheduling problems without considering the setup times. However, setup time plays significant role in many industries such as ceramic tile and paper cutting industries [53]. Hence, in this chapter we consider the flow shop scheduling problems with sequence dependent setup times. Moreover, researchers have proposed many bio-inspired metaheuristic algorithms recently. Firefly algorithm is one of the algorithms. We present the firefly algorithm to solve the flow shop scheduling problems to minimize the makespan, mean flow time, mean tardiness and number of tardy jobs. The remaining of the chapter is organised as follows. A brief review of literature is presented in Sect. 2. The problem definition is presented in Sect. 3. The proposed firefly algorithm is explained in detail in Sect. 4. Section 5 illustrates the computational results. Finally, the conclusions and future research opportunities are discussed in Sect. 6.

2 Literature Review

This section will briefly present the backgrounds necessary for the current study. It includes the flow shop scheduling problems with different objective functions and firefly algorithm.

2.1 Flow Shop Scheduling

The flow shop model was first proposed by Johnson [27] and many researchers addressed different types of scheduling problems for the past few decades. However, it has been reported that most of the researchers considered makespan minimization as the objective function [23] and very few of them only considered the setup times. Corwin and Esogbue [11] first proposed a two machine flow shop scheduling problems with sequence dependent setup times. They developed a dynamic programming to minimize the makespan. They compared the dynamic programming approach with the branch-and-bound algorithm. Gupta and Darrow [20] developed four efficient approximate algorithms to solve the two-machine flow shop scheduling problems sequence dependent setup times to minimize the makespan. Srikar and Ghosh [58] developed a mixed integer linear programming model for the flow shop scheduling problems with sequence dependent setup times. Osman and Potts [41] proposed a simulated annealing (SA) algorithm for solving the permutation flow-shop scheduling problems to minimize the maximum completion time. Rios-Mercado and Bard [51] addressed two new heuristics for solving the flow shop scheduling problems with sequence-dependent setup time to minimize the makespan. T'kindt et al. [60] applied the ant colony optimization (ACO) algorithm to solve a 2-machine bicriteria flow shop scheduling problem with the objective of minimizing both the total completion time and the makespan criteria.

The proposed algorithm heuristic also used the features of SA and local search algorithms. Computational experiments showed the effectiveness of the proposed algorithm.

Rajendran and Ziegler [49] proposed two ACO algorithms for permutation flow shop scheduling problems to minimize the makespan and total flow time. They tested the performance of the proposed algorithms with the benchmark problems addressed in the literature. França et al. [16] proposed two evolutionary algorithms namely genetic algorithm (GA) and a memetic algorithm (MA) with local search for solving flowshop scheduling problems with sequence dependent family setup times to obtain optimal schedule. Ravindran et al. [50] proposed three different heuristics to minimize the makespan and total flow time in a flow shop scheduling environment. The effectiveness of the heuristics was tested with the benchmark problems addressed in the literature. Ruiz et al. [53] addressed the different variants of GA to solve the permutation flowshop scheduling problems with sequence dependent setup times to minimize the makespan. They calibrated the parameters and operators of the GA by means of Design of Experiments. They tested the proposed algorithms with the benchmark problems addressed in the literature and proved that the proposed algorithms were superior. Liao et al. [32] addressed a discrete particle swarm optimization (DPSO) algorithm for solving the flow shop scheduling problems. A local search was also incorporated into the proposed algorithm. Computational results showed that the proposed PSO algorithm was very competitive. Marichelvam [34] addressed an improved hybrid cuckoo search algorithm to minimise the makespan in FSSPs. Chowdhury et al. [9] proposed a novel GA to solve the blocking flow shop problems to minimize the makespan.

Ignall and Schrage [25] applied the branch-and-bound algorithm to solve the flow shop scheduling problems to minimize the flow time. Rajendran [46] developed a heuristic algorithm for solving flow shop scheduling problems to minimize the total flowtime. Vempati et al. [67] developed an effective heuristic to solve the flow shop scheduling problems to minimise the total flow time. Neppalli et al. [40] developed two GA based approaches for solving the two-stage bicriteria flow shop scheduling problems. The objective was minimization of total flow time and makespan. Computational experiments showed that the proposed algorithms were effective. Rajendran and Ziegler [47] developed a new heuristic to minimize the sum of weighted flow time in a flow shop environment with sequence-dependent setup times of jobs. Extensive computational experiments showed that the proposed heuristic was faster and more effective than other heuristics. Gupta et al. [21] proposed a tabu search algorithm to minimise the total flow time in a flow shop environment. Gupta et al. [22] proposed several polynomial heuristic solution algorithms to solve the two-machine flow shop scheduling problems to minimize the total flow time and makespan. Tang and Liu [59] developed modified version of GA to solve the flow shop scheduling problems to minimise the mean flow time. Varadharajan and Rajendran [66] developed a multi-objective SA (MOSA) algorithm for solving the flow shop scheduling problems to minimise the makespan and total flow time. Nagano and Moccellin [38] proposed a constructive heuristics to minimise the mean flow time in a flow shop environment. Tasgetiren et al. [61]

applied the PSO algorithm to solve the flow shop scheduling problems for minimizing the makespan and total flow time. Yagmahan and Yenisey [71] addressed an ACO algorithm for solving the multi-objective flow shop scheduling problems. They considered minimization of makespan, total flow time and total machine idle time as the objective function. They compared the performance of the proposed algorithm with other multi-objective heuristics. Computational results showed that proposed algorithm was more effective and better than other methods compared. Dong et al. [14] suggested an iterated local search algorithm to minimise the total flow time in the flow shops. A distribution algorithm was proposed by Jarboui et al. [26] for minimizing the total flow time.

Zhang et al. [72] presented a hybrid GA for solving the flow shop scheduling problems with total flow time objective. Chakraborty and Turvey [7] addressed a differential evolution (DE) algorithm and Czapiński [12] addressed a parallel SA algorithm with genetic enhancement for solving the flow shop problems with flow time objective. A mathematical programming model was developed by Salmasi et al. [54] for minimizing total flow time of the flow shop with sequence dependent group scheduling. They proposed a tabu search algorithm and a hybrid ACO algorithm to solve the problem. Tseng and Lin [64] proposed a genetic local search algorithm for minimizing the total flow time. A discrete harmony search algorithm was developed for solving the flow shop scheduling problems to minimise the total flow time by Gao et al. [18]. Tasgetiren et al. [62] addressed a discrete artificial bee colony (ABC) algorithm to minimise the total flow time in permutation flow shops. Two constructive heuristics were developed by Gao et al. [19] to solve the no-wait flow shop scheduling problems to minimize the total flow time.

Simons [57] proposed several decomposition methods to solve the reentrant flow shop scheduling problems with sequence dependent setup times to minimize maximum lateness. Kim [31] also developed a branch-and-bound algorithm to minimize the total tardiness in a permutation flow shop environment. Murata et al. [37] developed a multi-objective GA (MOGA) to solve the flow shop scheduling problems to minimize the makespan, total tardiness and total flow time. Parthasarathy and Rajendran [43] also proposed the SA algorithm to solve the flow shop scheduling problems with sequence-dependent setup times to minimize mean weighted tardiness. They considered a drill-bit manufacturing industry. Computational results revealed that the proposed heuristic was better than many other algorithms. Armentano and Ronconi [1] proposed a tabu search based heuristic for solving the flow shop scheduling problems to minimize total tardiness. Rajendran and Ziegler [48] proposed heuristics to solve the flow shop scheduling problems with sequence-dependent setup times to minimize the sum of weighted flowtime and weighted tardiness of jobs.

Arroyo and Armentano [2] developed a genetic local search for solving the multi-objective flow shop scheduling problems. The algorithm was applied to the flowshop scheduling problem for the following two pairs of objectives: (i) makespan and maximum tardiness; (ii) makespan and total tardiness. The performance of the proposed algorithm was compared with two multi-objective genetic local search algorithms proposed in the literature. Computational results showed that the proposed algorithm was better than other algorithms. Rahimi-Vahed and Mirghorbani [45] developed a multi-objective PSO (MOPSO) algorithm for solving the flow shop scheduling problems to minimize the weighted mean completion time and weighted mean tardiness. The computational results showed that the proposed algorithm was better than the GA. Ruiz and Stützle [52] developed two new iterated greedy heuristics to solve the flowshop scheduling problems with sequence dependent setup times. Minimization of makespan and weighted tardiness were the objectives considered by them. Tavakkoli-Moghaddam et al. [63] addressed a hybrid multi-objective algorithm based on the features of a biological immune system (IS) and bacterial optimization (BO) to find Pareto optimal solutions for solving the multi-objective no-wait flow shop scheduling problems to minimize the weighted mean completion time and weighted mean tardiness. They compared the performance of the proposed algorithm against five different multi-objective evolutionary algorithms addressed in the literature and proved that the proposed algorithm was efficient. Naderi et al. [39] presented an electromagnetism-like mechanism and SA algorithms for flow shop scheduling problems for minimizing the total weighted tardiness and makespan. Pan et al. [42] presented a discrete ABC algorithm to solve the lot-streaming flow shop scheduling problems with the criterion of total weighted earliness and tardiness. They used the dispatching rules to construct the initial population. A simple and effective local search approach was also used by them. Khalili and Tavakkoli-Moghaddam [30] presented a new multi-objective electromagnetism algorithm (MOEM) to solve the bi-objective flowshop scheduling problems. The objective was to minimize the makespan and total weighted tardiness. They considered the transportation times between machines. They also applied the SA algorithm to solve the given problem. They conducted the computational experiments and proved that the proposed MOEM provided better results. Boxma and Forst [6] proposed heuristics to minimize the expected weighted number of tardy jobs in a stochastic flow shops.

2.2 Firefly Algorithm

Firefly algorithm (FA) is one of the recently developed meta-heuristic algorithms by Yang [70]. Sayadia et al. [55] presented a new discrete firefly algorithm (DFA) metaheuristic to minimize the makespan for the permutation flow shop scheduling problems. Computational results indicated that the proposed DFA was better than the ACO algorithm for the well known benchmark problems reported in the literature. Banati and Bajaj [4] presented a new feature selection approach by combining the rough set theory with the FA. The FA was applied by Gandomi et al. [17] to solve the mixed continuous/discrete structural optimisation problems. Kazemzadeh and Kazemzadeh [28] proposed an improved FA to solve the structural optimisation problems. Liu and Ye [33] solved the permutation flow shop scheduling problem by a FA to minimize the makespan. Computational experiments proved the efficiency of the proposed FA.

The FA has been applied to solve the clustering problems by Senthilnath et al. [56]. Chandrasekaran and Simon [8] proposed a binary real coded FA to solve the unit commitment problem. Coelho and Mariani [10] proposed the FA to solve the multivariable PID controller tuning problems. Dekhici et al. [13] applied the FA for economic power dispatching problems with pollutants emission. The FA was applied for vector quantization in image compression problems by Horng [24]. The job shop scheduling problems have been solved using the FA by Khadwilard et al. [29]. They also investigated the different parameters for the proposed algorithm and compared the performance with different parameters. The FA was applied for solving the economic load dispatching problems by Yang et al. [69].

An efficient FA was presented by Miguel et al. [36] to simultaneously optimize the size, shape and topology of the truss structures. They proved the effectiveness of the proposed FA by solving the benchmark problems reported in the literature. Fister et al. [15] presented a comprehensive review of FAs. Yang [68] proposed a multi-objective FA for continuous optimisation problems. Vahedi Nouri et al. [65] proposed a hybrid algorithm based on firefly and SA algorithms for solving the flow shop scheduling problems. The objective was to minimize the sum of tardiness costs and maintenance costs. A mixed integer linear programming model was proposed to formulate the problem. Marichelvam et al. [35] addressed a DFA for solving the hybrid flow shop scheduling problems to minimise the makespan and mean flow time.

3 Problem Definition

Flow shop scheduling environment consists of bank of m machines in series and n jobs are to be scheduled. Each job should be processed on the machines in a particular sequence. A job is first processed on machine 1, then on machine 2 and finally completed on machine m. The processing time of the jobs are known in advance, fixed and nonnegative. It is expected that the jobs are available at time zero. Each machine can processes only one job at a time and each job can be processed on only one machine at a time. The processing of each job cannot be interrupted, that is, preemption is not allowed. It is also assumed that the machines are available for the entire scheduling period (no machine breakdown). Minimization of makespan, mean flow time, mean tardiness and number of tardy jobs are the objective functions considered. The flow shop scheduling environment is given in Fig. 1 [34].

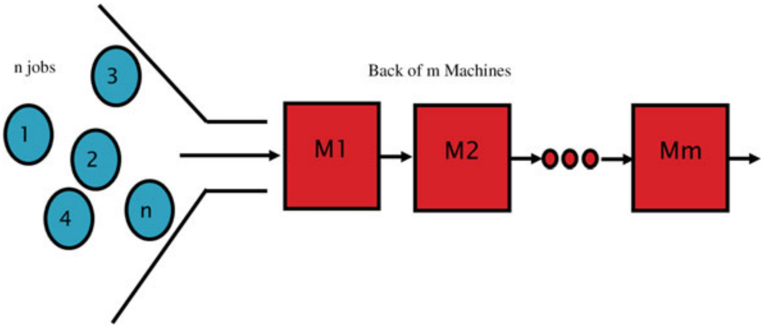

Fig. 1 Layout of a flow shop environment

3.1 Makespan (C_{max})

Makespan is the completion time of the last job to leave the system production system. Minimizing makespan would lead to the maximization of the resource utilisation and the throughput of a production system.

3.2 Mean Flow Time ($\bar{f}$)

Mean flow time is defined as the average time spent by the jobs in the production system. It is one of the most important performance measures. Mean flow time can help to effective utilization of resources, rapid turn-around of jobs, and minimization of work-in-process inventory costs.

3.3 Mean Tardiness ($\bar{T}$)

The tardiness is defined as the lateness of a job if it fails to meet its due date, or zero otherwise. Tardiness is associated with the service quality and customer satisfaction.

3.4 Number of Tardy Jobs (N_T)

This performance measure represents how many jobs are delayed in satisfying the due date. The detailed mathematical model for makespan, mean flow time, mean tardiness and number of tardy jobs can be found in [3].

3.5 Objective Function

The objective of this chapter is to minimize the weighted sum of makespan, mean flow time, mean tardiness and number of tardy jobs.

$$Z = w_1 C_{\max} + w_2 \bar{f} + w_3 \bar{T} + w_4 N_T \tag{1}$$

where w_1, w_2, w_3 and w_4 are the weight values of the objective functions and $w_1 \geq 0,\ w_2 \geq 0,\ w_3 \geq 0$ and $w_4 \geq 0$.

$$w_1 + w_2 + w_3 + w_4 = 1 \tag{2}$$

4 Firefly Algorithm

Firefly algorithm (FA) is a nature- inspired meta-heuristic algorithm. The FA is inspired by the social behavior of fireflies. Fireflies may also be called as lighting bugs. There are about two thousand firefly species in the world. Most of the firefly species produce short and rhythmic flashes. The pattern of flashes is unique for a particular species. A firefly's flash is mainly act as a signal system to attract mating partners and to attract potential prey. Flashes also serve as a protective warning mechanism. The following three idealized rules are considered for describing the FA [70].

1. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex
2. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move toward the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly
3. The brightness of a firefly is affected or determined by the landscape of the objective function. For a maximization problem, the brightness may be proportional to the objective function value. For the minimization problem the brightness may be the reciprocal of the objective function value.

The Pseudo code of the FA is given in Fig. 2.

4.1 Attractiveness of a Firefly

The attractiveness of a firefly is determined by its light intensity. The attractiveness may be calculated by using the Eq. (3).

Fig. 2 Pseudo code of the firefly algorithm

```
Objective function f(x), x = (x_1, ..., x_d)^T
Generate initial population of fireflies x_i (i = 1, 2,..., n)
Light intensity I_i at x_i is determined by f(x_i)
Define light absorption coefficient γ
While (t <MaxGeneration)
for i = 1 : n all n fireflies
for j = 1 : i all n fireflies
if (I_j > I_i ), Move firefly i towards j in d-dimension; end if
Attractiveness varies with distance r via exp [–γr^2]
Evaluate new solutions and update light intensity
end for j
end for i
Rank the fireflies and find the current best
end while
Postprocess results and visualization
```

$$\beta(\rho) = \beta_0 e^{-\gamma r^2} \tag{3}$$

where β is the attractiveness of a firefly and γ is the light absorption coefficient.

4.2 Distance Between Two Fireflies

The distance between any two fireflies k and l at X_k and X_l is the Cartesian distance using the Eq. (4).

$$r_{kl} = \|X_k - X_l\| = \sqrt{\sum_{k=1}^{d} (X_{k,o} - X_{l,o})^2} \tag{4}$$

4.3 Movement of a Firefly

The movement of a firefly k that is attracted to another more attractive firefly l is determined by the Eq. (5).

$$X_k = X_k + \beta_0 \mathrm{e}^{-\gamma r^2 kl}(X_l - X_k) + \alpha\left(rand - \frac{1}{2}\right) \tag{5}$$

where α is the randomization parameter and rand is a number uniformly drawn from the interval [0, 1].

4.4 Discrete Firefly Algorithm (DFA)

The FA has been originally developed for solving the continuous optimization problems. The FA cannot be applied directly to solve the discrete optimization problems. In this chapter, the smallest position value (SPV) rule described by Bean [5] is used to enable the continuous FA to be applied to solve the discrete HFS scheduling problems. For this, a discrete firefly algorithm (DFA) is proposed. More detail about the DFA can be found in [35].

4.5 Implementation of the DFA for Flow Shop Scheduling Problems

This section illustrates how the DFA is applied for solving the flow shop scheduling problems.

4.5.1 Solution Representation

Solution representation is one of the most important issues in designing a DFA. The solution search space consists of n dimensions as n number of jobs is considered in this book chapter. Each dimension represents a job. The vector $X_i^t = (X_{i1}^t, X_{i2}^t, \ldots, X_{in}^t)$ represents the continuous position values of fireflies in the search space. The SPV rule is used to convert the continuous position values of the fireflies to the discrete job permutation. The solution representation of a firefly with 6 jobs is illustrated in Table 1

The smallest position value is $x_{i4}^t = 0.07$and the dimension $j = 4$ is assigned to be the first job in the permutation according to the SPV rule. The second smallest position value is $x_{i3}^t = 0.22$ and the dimension $j = 3$ is assigned to be the second job in the permutation. Similarly, all the jobs are assigned in the permutation.

4.5.2 Population Initialization

In most of the meta-heuristics, the initial population is generated at random. In the DFA the initial population is also generated at random. The continuous values of positions are generated randomly using a uniform random number between 0 and 1.

Table 1 Solution representation of a FA

	Dimension j						
	1	2	3	4	5	6	7
x_{ij}	0.83	0.94	0.22	0.07	0.74	0.61	0.96
jobs	5	6	2	1	4	3	7

4.5.3 Solution Updation

By using the permutation, each firefly is evaluated to determine the objective function value. The objective function value of each firefly is associated with the light intensity of the corresponding firefly. A firefly with less brightness is attracted and moved to a firefly with more brightness. The attractiveness of the firefly is determined using the Eq. (3). The distance between each two fireflies is determined by the Eq. (4). The SPV rule is applied to obtain the job permutation. The attractiveness is calculated for each firefly. Then, the movement of the firefly is determined by the Eq. (5) depending on the attractiveness of the firefly. The above steps are repeated until the termination criterion is met.

5 Computational Results

The proposed algorithm was coded in C++ and run on a PC with an Intel Core Duo 2.4 GHz CPU, 2 GB RAM, running Windows XP. Simulation experiments are performed with different parameter settings to evaluate the performance of the FA. The factor levels for the design of experiments are given in Table 2.

Hence we conduct $3 \times 3 \times 1 \times 1 \times 1 \times 3 \times 3 \times 3 = 243$ experiments to evaluate the performance of the proposed algorithm. Each problem is tested with 20 replications. We compare the performance of the proposed discrete firefly algorithm with the genetic algorithm (GA), ant colony optimization (ACO) algorithm, cuckoo search (CS), particle swarm optimization (PSO) and the simulated annealing (SA)

Table 2 Factor levels for the design of experiments

Sl. No.	Factors	Levels
1	Number of jobs	20, 50 and 100
2	Number of machines	2, 5 and 10
3	Processing time distribution	Uniform (1–100)
4	Setup times	Uniform (0–10)
5	Due date	0.5–1.2 times processing time
6	Attractiveness of a firefly β_0	0.0 (low)
		0.5 (medium)
		1.0 (high)
7	Light absorption coefficient γ	0.5 (low)
		0.75 (medium)
		1.0 (high)
8	Randomization parameter α	0.0 (low)
		0.5 (medium)
		1.0 (high)

Table 3 Result comparison of different algorithms

Sl. No.	Number of jobs	Number of machines	β_0	γ	α	MRDI					
						SA	ACO	GA	CS	PSO	DFA
1	20	2	0	0.50	0	8.00	7.85	6.62	2.21	4.23	0.00
2	20	2	0	0.75	0	8.35	7.25	6.69	2.77	4.41	0.00
3	20	2	0	1.00	0	8.95	7.46	6.54	2.07	4.62	0.00
4	20	2	0	0.50	0.50	8.15	7.70	6.51	2.39	4.89	0.00
5	20	2	0	0.75	0.50	8.26	7.48	6.74	2.19	4.60	0.00
6	20	2	0	1.00	0.50	8.55	7.05	6.85	2.91	4.51	0.00
7	20	2	0	0.50	1.00	8.42	7.76	6.83	2.92	4.97	0.00
8	20	2	0	0.75	1.00	8.14	7.80	6.43	2.36	4.38	0.00
9	20	2	0	1.00	1.00	8.62	7.31	6.53	2.18	4.92	0.00
10	20	5	0.50	0.50	0	8.94	7.33	6.88	2.13	4.06	0.00
11	20	5	0.50	0.75	0	8.45	7.56	6.68	2.98	4.81	0.00
12	20	5	0.50	1.00	0	8.32	7.91	6.34	2.34	4.63	0.00
13	20	5	0.50	0.50	0.50	8.28	7.24	6.29	2.56	4.76	0.00
14	20	5	0.50	0.75	0.50	8.56	7.36	6.82	2.82	4.12	0.00
15	20	5	0.50	1.00	0.50	8.72	7.42	6.18	2.64	4.14	0.00
16	20	5	0.50	0.50	1.00	8.16	7.31	6.54	2.57	4.48	0.00
17	20	5	0.50	0.75	1.00	8.43	7.52	6.62	2.73	4.51	0.00
18	20	5	0.50	1.00	1.00	8.32	7.75	6.41	2.32	4.78	0.00
19	20	10	1.00	0.50	0	8.46	7.33	6.50	2.56	4.51	0.00
20	20	10	1.00	0.75	0	8.45	7.50	6.39	2.72	4.52	0.00
21	20	10	1.00	1.00	0	8.24	7.70	6.24	2.82	4.55	0.00
22	20	10	1.00	0.50	0.50	8.56	7.63	6.63	2.68	4.63	0.00
23	20	10	1.00	0.75	0.50	8.46	7.29	6.61	2.64	4.53	0.00

(continued)

Table 3 (continued)

Sl. No.	Number of jobs	Number of machines	β_0	γ	α	MRDI					
						SA	ACO	GA	CS	PSO	DFA
24	20	10	1.00	1.00	0.50	8.48	7.58	6.41	2.34	4.62	0.00
25	20	10	1.00	0.50	1.00	8.53	7.91	6.59	2.48	4.75	0.00
26	20	10	1.00	0.75	1.00	8.63	7.57	6.64	2.68	4.60	0.00
27	20	10	1.00	1.00	1.00	8.64	7.33	6.67	2.70	4.62	0.00
28	50	2	0	0.50	0	8.56	7.29	6.38	2.57	4.48	0.00
29	50	2	0	0.75	0	8.43	7.69	6.55	2.46	4.61	0.00
30	50	2	0	1.00	0	8.36	7.45	6.72	2.92	4.68	0.00
31	50	2	0	0.50	0.50	8.56	7.66	6.59	2.68	4.45	0.00
32	50	2	0	0.75	0.50	8.08	7.94	6.51	2.67	4.59	0.00
33	50	2	0	1.00	0.50	8.12	7.35	6.40	2.48	4.59	0.00
34	50	2	0	0.50	1.00	8.32	7.74	6.38	2.45	4.57	0.00
35	50	2	0	0.75	1.00	8.46	7.88	6.41	2.37	4.59	0.00
36	50	2	0	1.00	1.00	8.36	7.33	6.59	2.94	4.55	0.00
37	50	5	0.50	0.50	0	8.56	7.28	6.34	2.69	4.66	0.00
38	50	5	0.50	0.75	0	8.00	7.71	6.45	2.54	4.56	0.00
39	50	5	0.50	1.00	0	8.35	7.60	6.52	2.73	4.62	0.00
40	50	5	0.50	0.50	0.50	8.95	7.43	6.55	2.68	4.70	0.00
41	50	5	0.50	0.75	0.50	8.15	7.21	6.54	2.54	4.67	0.00
42	50	5	0.50	1.00	0.50	8.26	7.83	6.67	2.63	4.44	0.00
43	50	5	0.50	0.50	1.00	8.46	7.78	6.62	2.58	4.56	0.00
44	50	5	0.50	0.75	1.00	8.45	7.13	6.72	2.63	4.81	0.00
45	50	5	0.50	1.00	1.00	8.24	7.89	6.54	2.56	4.42	0.00
46	50	10	1.00	0.50	0	8.00	7.58	6.54	2.83	4.58	0.00

(continued)

Table 3 (continued)

Sl. No.	Number of jobs	Number of machines	β_0	γ	α	MRDI					
						SA	ACO	GA	CS	PSO	DFA
47	50	10	1.00	0.75	0	8.35	7.52	6.50	2.67	4.59	0.00
48	50	10	1.00	1.00	0	8.95	7.64	6.40	2.48	4.60	0.00
49	50	10	1.00	0.50	0.50	8.15	7.20	6.73	2.78	4.54	0.00
50	50	10	1.00	0.75	0.50	8.26	7.84	6.50	2.48	4.57	0.00
51	50	10	1.00	1.00	0.50	8.43	7.53	6.45	2.69	4.48	0.00
52	100	10	1.00	0.50	1.00	8.32	7.82	6.40	2.78	4.37	0.00
53	100	10	1.00	0.75	1.00	8.46	7.18	6.63	2.82	4.65	0.00
54	100	10	1.00	1.00	1.00	8.45	7.40	6.54	2.49	4.65	0.00
55	100	2	0	0.50	0	8.24	7.34	6.53	2.68	4.49	0.00
56	100	2	0	0.75	0	8.00	7.23	6.67	2.58	4.65	0.00
57	100	2	0	1.00	0	8.53	7.64	6.56	2.76	4.66	0.00
58	100	2	0	0.50	0.50	8.59	7.54	6.65	2.97	4.64	0.00
59	100	2	0	0.75	0.50	8.21	7.42	6.36	2.67	4.46	0.00
60	100	2	0	1.00	0.50	8.62	7.46	6.53	2.91	4.63	0.00
61	100	2	0	0.50	1.00	8.55	7.64	6.62	2.68	4.77	0.00
62	100	2	0	0.75	1.00	8.42	7.84	6.34	2.59	4.62	0.00
63	100	2	0	1.00	1.00	8.14	7.43	6.54	2.67	4.58	0.00
64	100	5	0.50	0.50	0	8.62	7.36	6.59	2.46	4.49	0.00
65	100	5	0.50	0.75	0	8.94	7.46	6.47	2.73	4.43	0.00
66	100	5	0.50	1.00	0	8.46	7.62	6.54	2.91	4.51	0.00
67	100	5	0.50	0.50	0.50	8.45	7.43	6.54	2.67	4.71	0.00
68	100	5	0.50	0.75	0.50	8.24	7.36	6.64	2.68	4.45	0.00
69	100	5	0.50	1.00	0.50	8.00	7.56	6.46	2.74	4.51	0.00

(continued)

Table 3 (continued)

Sl. No.	Number of jobs	Number of machines	β_0	γ	α	MRDI					
						SA	ACO	GA	CS	PSO	DFA
70	100	5	0.50	0.50	1.00	8.35	7.28	6.58	2.68	4.61	0.00
71	100	5	0.50	0.75	1.00	8.95	7.12	6.74	2.56	4.53	0.00
72	100	5	0.50	1.00	1.00	8.15	7.32	6.64	2.74	4.44	0.00
73	100	10	1.00	0.50	0	8.26	7.46	6.55	2.82	4.59	0.00
74	100	10	1.00	0.75	0	8.55	7.36	6.50	2.72	4.67	0.00
75	100	10	1.00	1.00	0	8.42	7.56	6.61	2.74	4.53	0.00
76	100	10	1.00	0.50	0.50	8.14	7.82	6.48	2.68	4.59	0.00
77	100	10	1.00	0.75	0.50	8.62	7.64	6.65	2.73	4.46	0.00
78	100	10	1.00	1.00	0.50	8.94	7.20	6.57	2.77	4.57	0.00
79	100	10	1.00	0.50	1.00	8.54	7.84	6.34	2.74	4.77	0.00
80	100	10	1.00	0.75	1.00	8.42	7.53	6.58	2.68	4.43	0.00
81	100	10	1.00	1.00	1.00	8.12	7.82	6.70	2.67	4.58	0.00

algorithms addressed in the literature. Mean Relative Deviation Index (MRDI) is used as a performance measure to compare the performance of different algorithms. MRDI is calculated as given below:

$$MRDI = \sum_{l=1}^{R} \frac{|Z^{*} - Z_{META}|}{Z^{*}} \times 100/R \quad (6)$$

Where,

Z^{*}	best objective function value
Z_{META}	objective function value obtained by different metaheuristic algorithms
R	number of runs (20)

Lower value of MRDI value indicates the better performance of the algorithm. The comparison results of some sample problems are presented in Table 3.

From the result table, it can be easily concluded that the proposed DFA is better than many other algorithms addressed in the literature.

6 Conclusions

A discrete firefly algorithm is presented in this chapter to minimize the weighted sum of makespan, mean flow time, mean tardiness and number of tardy jobs for flow shop scheduling problems. The proposed DFA has been tested over a set of random problem instances with different parameter settings and the results have been compared with other metaheuristics addressed in the literature. It is concluded that the DFA provides better results than many other metaheuristics. This work may be extended in many directions. The algorithm can also be applied to solve real industrial scheduling problems. The research may also be conducted for other types of scheduling problems. It would be interesting to conduct the computational experiments with several other parameters values to determine the optimal parameters of the firefly algorithm.

References

1. Armentano, V.A., Ronconi, D.P.: Tabu search for total tardiness minimization in flowshop scheduling problems. Comp. Oper. Res. **26**(3), 219–235 (1999)
2. Arroyo, J.E.C., Armentano, V.A.: Genetic local search for multi-objective flowshop scheduling problems. Eur. J. Oper. Res. **167**, 717–738 (2005)
3. Baker, K.R.: Introduction to sequencing and scheduling. Wiley, New York (1974)
4. Banati, H., Bajaj, M.: Firefly based feature selection approach. Int. J. Comp. Sci. Issues **8**(4), 473–480 (2011)

5. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA J. comput. **6**(2), 154–160 (1994)
6. Boxma, O.J., Forst, F.G.: Minimizing the expected weighted number of tardy jobs in stochastic flow shops. Oper. Res. Lett. **5**(3), 119–126 (1986)
7. Chakraborty, U.K., Turvey, K.P.: Floating-point to integer mapping schemes in differential evolution for permutation flow shop scheduling. Int. J. Bio-Inspired Comput. **2**(3), 183–204 (2010)
8. Chandrasekaran, K., Simon, S.P.: Network and reliability constrained unit commitment problem using binary real coded firefly algorithm. Int. J. Electr. Power Energy Syst. **43**(1), 921–932 (2012)
9. Chowdhury, A., Ghosh, A., Sinha, S., Das, S., Ghosh, A.: A novel genetic algorithm to solve travelling salesman problem and blocking flow shop scheduling problem. Int. J. Bio-Inspired Comput. **5**(5), 303–314 (2013)
10. Coelho, L.D.S., Mariani, V.C.: Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. Comput. Math. Appl. **64**(8), 2371–2382 (2012)
11. Corwin, B.D., Esogbue, A.O.: Two machine flow shop scheduling problems with sequence dependent setup times: a dynamic programming approach. Naval Res. Logistics Q. **21**(3), 515–524 (1974)
12. Czapiński, M.: Parallel simulated annealing with genetic enhancement for flowshop problem with C_{sum}. Comput. Ind. Eng. **59**(4), 778–785 (2010)
13. Dekhici, L., Borne, P., Khaled, B.: Firefly algorithm for economic power dispatching with pollutants emission. Informatica Economică **16**(2), 45–57 (2012)
14. Dong, X., Huang, H., Chen, P.: An iterated local search algorithm for the permutation flowshop problem with total flowtime criterion. Comput. Oper. Res. **36**(5), 1664–1669 (2009)
15. Fister, I., Fister Jr, I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm Evol. Comput. **13**, 34–46 (2013)
16. França, P.M., Gupta, J.N., Mendes, A.S., Moscato, P., Veltink, K.J.: Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. Comput. Ind. Eng. **48**(3), 491–506 (2005)
17. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Mixed variable structural optimization using firefly algorithm. Comput. Struct. **89**(23), 2325–2336 (2011)
18. Gao, K.Z., Pan, Q.K., Li, J.Q.: Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion. Int. J. Adv. Manuf. Technol. **56**(5–8), 683–692 (2011)
19. Gao, K., Pan, Q., Suganthan, P.N., Li, J.: Effective heuristics for the no-wait flow shop scheduling problem with total flow time minimization. Int. J. Adv. Manuf. Technol. **66**(9–12), 1563–1572 (2013)
20. Gupta, J.N., Darrow, W.P.: The two-machine sequence dependent flowshop scheduling problem. Eur. J. Oper. Res. **24**(3), 439–446 (1986)
21. Gupta, J.N.D., Chen, C.L., Yap, L.Y., Deshmukh, H.: Designing a tabu search algorithm to minimize total flow time in a flow shop. Arab. J. Sci. Eng. **25**(1), 79–94 (2000)
22. Gupta, J.N., Neppalli, V.R., Werner, F.: Minimizing total flow time in a two-machine flowshop problem with minimum makespan. Int. J. Prod. Econ. **69**(3), 323–338 (2001)
23. Gupta, J.N., Stafford Jr, E.F.: Flowshop scheduling research after five decades. Eur. J. Oper. Res. **169**(3), 699–711 (2006)
24. Horng, M.H.: Vector quantization using the firefly algorithm for image compression. Expert Syst. Appl. **39**(1), 1078–1091 (2012)
25. Ignall, E., Schrage, L.: Application of the branch and bound technique to some flow-shop scheduling problems. Oper. Res. **13**(3), 400–412 (1965)
26. Jarboui, B., Eddaly, M., Siarry, P.: An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. Comput. Oper. Res. **36**(9), 2638–2646 (2009)
27. Johnson, S.M.: Optimal two and three stage production schedules with setup times included. Naval Res. Logistics Q. **1**(1), 61–68 (1954)

28. Kazemzadeh, A.S., Kazemzadeh, A.S.: Optimum design of structures using an improved firefly algorithm. Int. J. Optim. Civ. Eng. **2**, 327–340 (2011)
29. Khadwilard, A., Chansombat, S., Thepphakorn, T., Thapatsuwan, P., Chainat, W., Pongcharoen, P.: Application of firefly algorithm and its parameter setting for job shop scheduling. J. Ind. Technol. **8**, 49–58 (2012)
30. Khalili, M., Tavakkoli-Moghaddam, R.: A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. J. Manuf. Syst. **31**(2), 232–239 (2012)
31. Kim, Y.D.: Minimizing total tardiness in permutation flowshops. Eur. J. Oper. Res. **85**(3), 541–555 (1995)
32. Liao, C.J., Tseng, C.T., Luarn, P.: A discrete version of particle swarm optimization for flowshop scheduling problems. Comput. Oper. Res. **34**(10), 3099–3111 (2007)
33. Liu, C.P., Ye, C.M.: Solving Permutation Flow Shop Scheduling Problem by firefly Algorithm. Ind. Eng. Manage. **17**(3), 56–59 (2012)
34. Marichelvam, M.K.: An improved hybrid Cuckoo Search (IHCS) metaheuristics algorithm for permutation flow shop scheduling problems. Int. J. Bio-Inspired Comput. **4**(4), 200–205 (2012)
35. Marichelvam, M.K., Prabaharan, T., Yang, X.S.: A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. IEEE Trans. Evol. Comput. **18**(2), 301–305 (2014)
36. Miguel, L.F.F., Lopez, R.H., Miguel, L.F.F.: Multimodal size, shape, and topology optimisation of truss structures using the firefly algorithm. Adv. Eng. Softw. **56**, 23–37 (2013)
37. Murata, T., Ishibuchi, H., Tanaka, H.: Multi-objective genetic algorithm and its applications to flowshop scheduling. Comput. Ind. Eng. **30**(4), 957–968 (1996)
38. Nagano, M.S., Moccellin, J.V.: Reducing mean flow time in permutation flow shop. J. Oper. Res. Soc. **59**(7), 939–945 (2008)
39. Naderi, B., Tavakkoli-Moghaddam, R., Khalili, M.: Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan. Knowl.-Based Syst. **23**(2), 77–85 (2010)
40. Neppalli, V.R., Chen, C.L., Gupta, J.N.: Genetic algorithms for the two-stage bicriteria flowshop problem. Eur. J. Oper. Res. **95**(2), 356–373 (1996)
41. Osman, I.H., Potts, C.N.: Simulated annealing for permutation flow-shop scheduling. Omega **17**(6), 551–557 (1989)
42. Pan, Q.K., Fatih Tasgetiren, M., Suganthan, P.N., Chua, T.J.: A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf. Sci. **181**(12), 2455–2468 (2011)
43. Parthasarathy, S., Rajendran, C.: An experimental evaluation of heuristics for scheduling in a real-life flowshop with sequence-dependent setup times of jobs. Int. J. Prod. Econ. **49**(3), 255–263 (1997)
44. Pinedo, M.: Scheduling: theory, algorithms, and systems. Prentice-Hall, New Jersey (2002)
45. Rahimi-Vahed, A.R., Mirghorbani, S.M.: A multi-objective particle swarm for a flow shop scheduling problem. J. Comb. Optim. **13**(1), 79–102 (2007)
46. Rajendran, C.: Heuristic algorithm for scheduling in a flowshop to minimize total flowtime. Int. J. Prod. Econ. **29**(1), 65–73 (1993)
47. Rajendran, C., Ziegler, H.: An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs. Eur. J. Oper. Res. **103**(1), 129–138 (1997)
48. Rajendran, C., Ziegler, H.: Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times. Eur. J. Oper. Res. **149**(3), 513–522 (2003)
49. Rajendran, C., Ziegler, H.: Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. Eur. J. Oper. Res. **155**(2), 426–438 (2004)
50. Ravindran, D., Selvakumar, S.J., Sivaraman, R., Haq, A.N.: Flow shop scheduling with multiple objective of minimizing makespan and total flow time. Int. J. Adv. Manuf. Technol. **25**(9–10), 1007–1012 (2005)

51. Rios-Mercado, R.Z., Bard, J.F.: Heuristics for the flow line problem with setup costs. Eur. J. Oper. Res. **110**(1), 76–98 (1998)
52. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. Eur. J. Oper. Res. **177**(3), 2033–2049 (2007)
53. Ruiz, R., Maroto, C., Alcaraz, J.: Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. Eur. J. Oper. Res. **165**(1), 34–54 (2005)
54. Salmasi, N., Logendran, R., Skandari, M.R.: Total flow time minimization in a flowshop sequence-dependent group scheduling problem. Comput. Oper. Res. **37**(1), 199–212 (2010)
55. Sayadi, M., Ramezanian, R., Ghaffari-Nasab, N.: A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. Int. J. Ind. Eng. Comput. **1**(1), 1–10 (2010)
56. Senthilnath, J., Omkar, S.N., Mani, V.: Clustering using firefly algorithm: performance study. Swarm Evol. Comput. **1**(3), 164–171 (2011)
57. Simons Jr, J.V.: Heuristics in flow shop scheduling with sequence dependent setup times. Omega **20**(2), 215–225 (1992)
58. Srikar, B.N., Ghosh, S.: A MILP model for the n-job, m-stage flowshop with sequence dependent set-up times. Int. J. Prod. Res. **24**(6), 1459–1474 (1986)
59. Tang, L., Liu, J.: A modified genetic algorithm for the flow shop sequencing problem to minimize mean flow time. J. Intell. Manuf. **13**(1), 61–67 (2002)
60. TT'kindt, V., Monmarché, N., Tercinet, F., Laügt, D.: An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. Eur. J. Oper. Res. **142**(2), 250–257 (2002)
61. Tasgetiren, M.F., Liang, Y.C., Sevkli, M., Gencyilmaz, G.: A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. Eur. J. Oper. Res. **177**(3), 1930–1947 (2007)
62. Tasgetiren, M.F., Pan, Q.K., Suganthan, P.N., Chen, A.H.: A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. Inf. Sci. **181**(16), 3459–3475 (2011)
63. Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., Mirzaei, A.H.: A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. Inf. Sci. **177**(22), 5072–5090 (2007)
64. Tseng, L.Y., Lin, Y.T.: A genetic local search algorithm for minimizing total flowtime in the permutation flowshop scheduling problem. Int. J. Prod. Econ. **127**(1), 121–128 (2010)
65. Vahedi Nouri, B., Fattahi, P., Ramezanian, R.: Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities. Int. J. Prod. Res. **51**(12), 3501–3515 (2013)
66. Varadharajan, T.K., Rajendran, C.: A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. Eur. J. Oper. Res. **167**(3), 772–795 (2005)
67. Vempati, V.S., Chen, C.L., Bullington, S.F.: An effective heuristic for flow shop problems with total flow time as criterion. Comput. Ind. Eng. **25**(1), 219–222 (1993)
68. Yang, X.S.: Multiobjective firefly algorithm for continuous optimization. Eng. Comput. **29**(2), 175–184 (2013)
69. Yang, X.S., Sadat Hosseini, S.S., Gandomi, A.H.: Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl. Softw. Comput. **12**(3), 1180–1186 (2012)
70. Yang, X.S.: Nature-Inspired metaheuristic algorithms. Luniver press, UK (2008)
71. Yagmahan, B., Yenisey, M.M.: Ant colony optimization for multi-objective flow shop scheduling problem. Comput. Ind. Eng. **54**(3), 411–420 (2008)
72. Zhang, Y., Li, X., Wang, Q.: Hybrid genetic algorithm for permutation flowshop scheduling problems with total flowtime minimization. Eur. J. Oper. Res. **196**(3), 869–876 (2009)