

Solving the Flexible Job Shop Scheduling Problem Using a Hybrid Artificial Bee Colony Algorithm



Rylan H Caldeira , A. Gnanavelbabu , and J. Joseph Solomon 

Abstract In this work, a hybrid artificial bee colony algorithm is proposed for solving the flexible job shop scheduling problem (FJSP) which is a classification of the classical job shop scheduling problem (JSP) considered to NP-hard in nature. In FJSP, an operation can be processed on a set of capable machines with different processing times, thereby dealing with a routing and sequencing problem. The objective considered is to minimize the makespan. The basic artificial bee colony (ABC) algorithm stresses on the balance between global exploration and local exploitation. However, the drawback of the basic ABC algorithm is that it converges prematurely and may get trapped in the local optima. Hence to improve its exploration capability in local space, it is hybridized using a Tabu search (TS) algorithm. At first, initial solutions are generated with certain quality and diversity as food sources using multiple strategies in combination. Crossover and mutation operations are carried out for machine assignment and operation sequencing separately generating new neighboring solutions. Lastly, a local search strategy based on TS is proposed to enhance the local search capability. Kacem's and Brandimarte's benchmark instances are used to compare the performance of the proposed approach to five other well-known algorithms in the literature. Experimental results revealed the superiority of the proposed approach in solving FJSP.

Keywords Flexible job shop scheduling · Artificial bee colony algorithm · Tabu search · Makespan

R. H. Caldeira · A. Gnanavelbabu (✉) · J. Joseph Solomon
Department of Industrial Engineering, CEG Campus, Anna University, Chennai,
Tamil Nadu 600025, India
e-mail: agbabu@annauniv.edu

© Springer Nature Singapore Pte Ltd. 2021
S. Vijayan et al. (eds.), *Trends in Manufacturing and Engineering Management*,
Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-15-4745-4_72

1 Introduction

The process of converting the resources into a finished or semi-finished product by adding value to it is called manufacturing. There are various types of manufacturing systems based on their size and activity. They are job shop, batch production, mass production, flow production, and assembly lines. This work deals with job shop since it is an extensively adopted type of production system. A job shop is a type of manufacturing process in which small batches of a variety of customized products are produced. JSP is the process of scheduling these jobs to machines. FJSP which is an extension of the JSP is an NP-hard combinatorial optimization problem. These problems are complex as each operation can be processed on more than one machine with different processing times. It consists of two sub-problems, routing, and scheduling. FJSP is usually solved by adopting a hierarchical or integrated approach. Brandimarte [1] proposed a hierarchical strategy for solving the scheduling problem by using TS. In recent times, hybrid meta-heuristics have become a popular tool for solving the FJSP as the hybridization of two or more meta-heuristics have proven to improve the performance of each meta-heuristic while eliminating the drawbacks of the other. ABC is one of the popular meta-heuristics. ABC provides a structured search framework that is characterized by individual improvement, cooperation, and competition of the population, and thus, it can perform global exploration efficiently (large-scale search). But the original ABC algorithm falls back in its ability to perform exploitation effectively by converging to a solution in earlier iterations, thus being incapable of exploring the search space effectively. In order to improve the local search capability of the ABC algorithm, a local search technique based on TS is introduced which greatly enhances the exploitation capability of the algorithm.

A beer froth ABC was proposed by Sharma et al. [2] in which to maintain proper harmony amid exploration and exploitation capabilities a beer froth phenomenon inspired position update was incorporated. A novel ABC based on the cosine similarity is proposed by Xiang et al. [3], a method based on cosine similarity for choosing a better neighbor individual to overcome the insufficiency of slow convergence in ABC. Levy flight ABC proposed by Sharma et al. [4] illustrated that there is a higher probability of skipping the optimal solution owing to large step sizes and a search strategy based on Levy flight was integrated with ABC which balances the exploitation and exploration capability. A hybrid ABC for FJSP proposed by Thammano and Phu-ang [5] where dispatching rules and the harmony search algorithm were used to create the initial solutions after which one of the two search techniques was randomly selected with a probability that is proportional to their fitness values. Simulated annealing algorithm or filter and fan algorithm was employed to escape from the local optimum. A hybrid ABC for the JSP proposed by Zhang et al. [6] aimed at minimizing the total weighted tardiness in JSP. Considering the high complexity, a novel ABC algorithm was proposed for solving the problem. A neighborhood property of the problem was discovered, and then, a tree search algorithm was devised to enhance the exploitation capability of ABC. An effective ABC for the FJSP proposed by Wang et al. [7] stresses the balance between global exploration and local

exploitation. A local search strategy based on critical path theory was proposed to enhance the local intensification capability of the onlooker bees.

A hybrid ABC for solving the FJSP proposed by Li et al. [8] considered the total flow time criterion, where TS heuristic is introduced to perform a local search for employee, onlooker, and scout bee phases. A best-so-far ABC proposed by Banharnsakun et al. [9] biases the solution direction toward the best-so-far solution rather than a neighboring solution as proposed in original ABC method and also uses the set theory to describe a mapping of their proposed method to the problem in the combinatorial optimization domain. Among different approaches, ABC algorithm proposed by Karaboga and Gorkemli [10] is a widely employed swarm intelligence algorithm for scheduling FJSP problems. ABC algorithm was first proposed to solve the multi-variable and multi-modal continuous functions. Localization and multi-objective evolutionary optimization for FJSP proposed by Kacem et al. [11] present two new approaches to jointly solve routing and scheduling by localization and then by evolutionary approach. Mehrabad and Fattahi [12] proposed a TS algorithm to solve the FJSP to minimize makespan. Hurink et al. [13] performed TS for solving the FJSP.

In this work, we propose a hybrid ABC with TS to improve the exploration capability in local space (exploitation). The proposed algorithm consists of the basic ABC phases, initialization, employee bee phase, and the scout bee phase. Here, solutions are initialized as two separate chromosomes to facilitate individual crossover. The two chromosomes are machine assignment and operation sequencing. Machine assignment follows the random rule, maximum time remaining rule, and most number of remaining operations rule, whereas operation sequencing follows the random rule, global minimum processing time rule, and work-time considered rule. These rules are adopted so that the population generated is diverse and feasible.

Employee bee phase initiates with the modified precedence operation crossover operator (MPOX) for crossover in operation sequencing. A uniform crossover is used for machine assignment. For half of the population chosen at random maximum workload reduction, mutation operator is used for modifying machine assignment. The mutation operator is used to enhance the exploration capability of the algorithm. The solution obtained is now introduced into a TS local search. This search performs an iterative search on the entire population to get new and better solutions in each step. TS is chosen as it prevents the local search from converging on a solution prematurely, thereby improving the local exploration capability of the algorithm. When the solution does not improve, a random solution is generated on behalf of the scout bee phase which is compared with the TS solution. A new better population is obtained from both of these solutions and taken for the next ABC iteration until the termination criteria are satisfied.

2 Problem Definition

The formulation of FJSSP is as follows: There is a set of ‘ m ’ machines $c = 1, 2, \dots, m$ and a set of ‘ n ’ jobs $a = 1, 2, \dots, n$. Each job ‘ a ’ consists of a sequence of ‘ k ’ operations. Processing of each operation O_{ab} ($b = 1, 2, \dots, k$) of job ‘ a ’ has to be done on single machine ‘ M_c ’ out of a set of given available machines M_{ab} (for $M_c \in M_{ab}, M_{ab} \in M$). Each operation needs a single machine picked from a set of capable machines. Further, FJSP requires to set starting and ending times of each operation. The FJSP needs to find both allotment and an operations sequence on the machines so that given criteria are satisfied. If there is at least one operation $M_{ab} \in M$, it is partial flexibility FJSP (P-FJSP), whereas if $M_{ab} = M$ for every operation, it is total flexibility FJSP (T-FJSP). The objective is to minimize the makespan.

3 Framework of the Proposed Approach

3.1 Artificial Bee Colony Algorithm

The ABC algorithm is a meta-heuristic designed by observing the foraging behavior of honey bee, introduced by Karaboga [1] in 2005. Here, all possible solutions to the optimization problem are considered as food sources, and the honey bees search to find the best solution. There are three types of bees in the population which are involved in finding the food source. They are

- **Employer bees.** For all employer bees, generate new candidate solutions and retain the candidate solutions possessing better objective function value.
- **Onlooker bee phase.** For all onlooker bees, select an employed bee randomly. Produce a new candidate solution. Compute fitness of individual, and if the fitness of the new solution is better than the existing solution, replace the older solution.
- **Scout bees.** If any source is exhausted, replace it by randomly generated solution by scout and memorize the best solution until stopping criteria are met. Old is substituted by an arbitrarily picked source within a specified region.

3.2 Solution Representation

The problem is solved hierarchically, and thus, solutions are represented as separate vectors each for operation sequencing and machine assignment which is termed as the two-vector solution representation (see Fig. 1). To guarantee an initial population with a certain quality, different heuristics are adopted to generate a population.

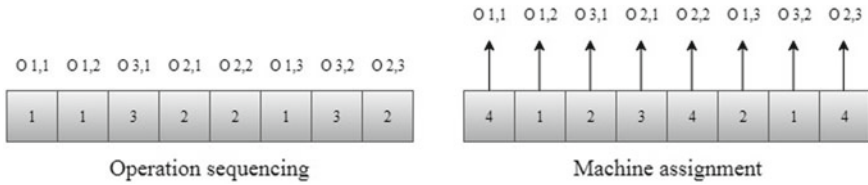


Fig. 1 Two-vector solution representation

For machine assignment, the following heuristics are employed:

- Global minimum processing time rule (20%);
- Workload considered rule (20%);
- Random rule (60%).

For operation sequencing, the following heuristics are employed:

- Random rule (80%);
- Most time remaining rule (10%);
- Most no. of operations remaining rule (10%).

3.3 Crossover and Mutation

In this phase, the two vectors are modified separately by crossover and mutation operators. Machine assignment vector undergoes a uniform crossover (see Fig. 3), where the new vector u_{new} is derived from the initial vector u_i and the vector with the best solution u' by generating a binary vector and then a maximum workload reduction mutation operator is used to reduce the assignment of the machine with maximum workload.

Operation sequencing vector undergoes a modified precedence operation crossover (MPOX) operation (see Fig. 2), where a random subset of half the jobs is generated, and they are given priority from one vector, whereas the other jobs are

Fig. 2 Modified precedence operation crossover

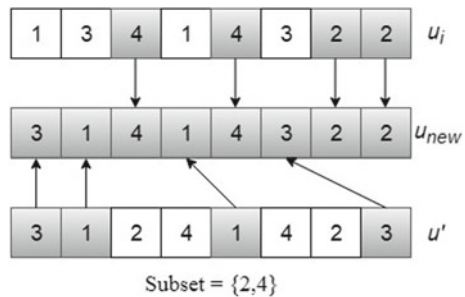
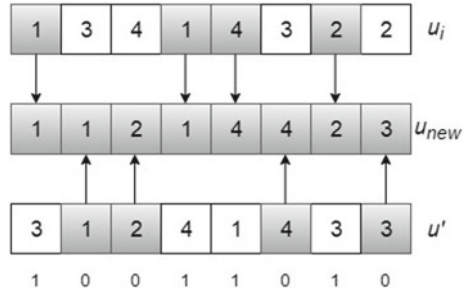


Fig. 3 Uniform crossover



assigned from another vector from left to right. Then, a maximum workload reduction operator is applied which identifies the machine with the maximum workload and re-assigns possible operations to other machines with lesser workloads. This acts as the mutation operator which is performed on only half of the population (Fig. 3).

3.4 Tabu Search

The TS is a local search heuristic that uses a neighborhood search procedure to iteratively move from one possible solution to x to another improved solution x' in the neighborhood of x until a terminating criterion is fulfilled. Local search techniques usually get stuck in less efficient areas and do not explore the search space completely. TS carefully explores the neighborhood of each solution as the search progresses. So to prevent the heuristic from returning to a recently visited solution, a short-term memory of recent moves within the search space is recorded, and this prevents future moves from repeating these recent moves. That is if a new potential solution has been previously visited within a short time period, it is marked as Tabu (forbidden) and the algorithm does not consider that solution. Thus, the solutions admitted to the new neighborhood, $N^*(x)$, are determined through the use of memory structures. Using these memory structures, the search progresses iteratively moves from the current solution x to an improved solution x' in $N^*(x)$.

These memory structures form a list known as the Tabu list which will contain a set of rules and banned solutions used to filter solutions that will be admitted to the neighborhood $N^*(x)$ to be explored by the search. So a Tabu list is a short-term set of the solutions that have been visited in the recent past (less than n iterations ago, where n is the number of previous solutions to be stored which is also called the Tabu tenure). So a Tabu list consists of solutions that have changed by the process of moving from one solution to another.

3.5 Proposed Hybrid ABC

The parameters for the algorithm like population size, number of iterations, and bee colony size are set. The initial population solution representation is done by using two-vector notations employing population initialization rules.

In the employee bee phase, the machine assignment vector undergoes uniform crossover and a maximum workload reduction mutation operation where the mutation is performed on only half of the population, whereas the operation sequencing vector undergoes a modified operation crossover operation. Then, for the whole modified population, a TS local search is performed which provides the best solution in every iteration without converging predominantly, thus improving the exploration capability of the ABC algorithm in local search space. The flowchart of the proposed hybrid ABC is illustrated (see Fig. 4).

4 Experimental Results

The proposed approach was coded in MATLAB software on an i3 processor with a speed of 1.8 GHz on a Windows 10 operating system. Computational experiments are conducted on all ten instances of Brandimarte [1] and five instances of Kacem et al. [11] since these instances are widely solved in literature and hence extensive comparison of solution methodologies can be carried out. Kacem's instances range from sizes of 4 jobs and 5 machines to 15 jobs and 10 machines with a single instance of 8 jobs and 8 machines with partial flexibility, while the remaining have total flexibility. Brandimarte's instances consist of problems ranging from sizes of 10 jobs and 6 machines to 20 jobs and 15 machines with medium flexibility.

From Table 1, it can be seen that the proposed hybrid approach obtains best solutions in ten out of fifteen benchmark instances. In comparison with other algorithms, it can be seen that the proposed algorithm outperformed the algorithm proposed by Xing et al. [14] in seven problems. The performance of the hybrid ABC was superior to TLBO proposed by Buddala et al. [15] in six problem instances. The proposed approach gave better results to two problems each compared to AIA proposed by Bagheri et al. [16] and HTSA proposed by Li et al. [17]. The proposed ABC outperformed BBO proposed by Rahmati et al. [18] in eight problem instances. The convergence of Brandimarte's MK09 problem is shown (see Fig. 5). Gantt chart obtained using the proposed approach is shown for Kacem's 10 * 15 problem instance (see Fig. 6).

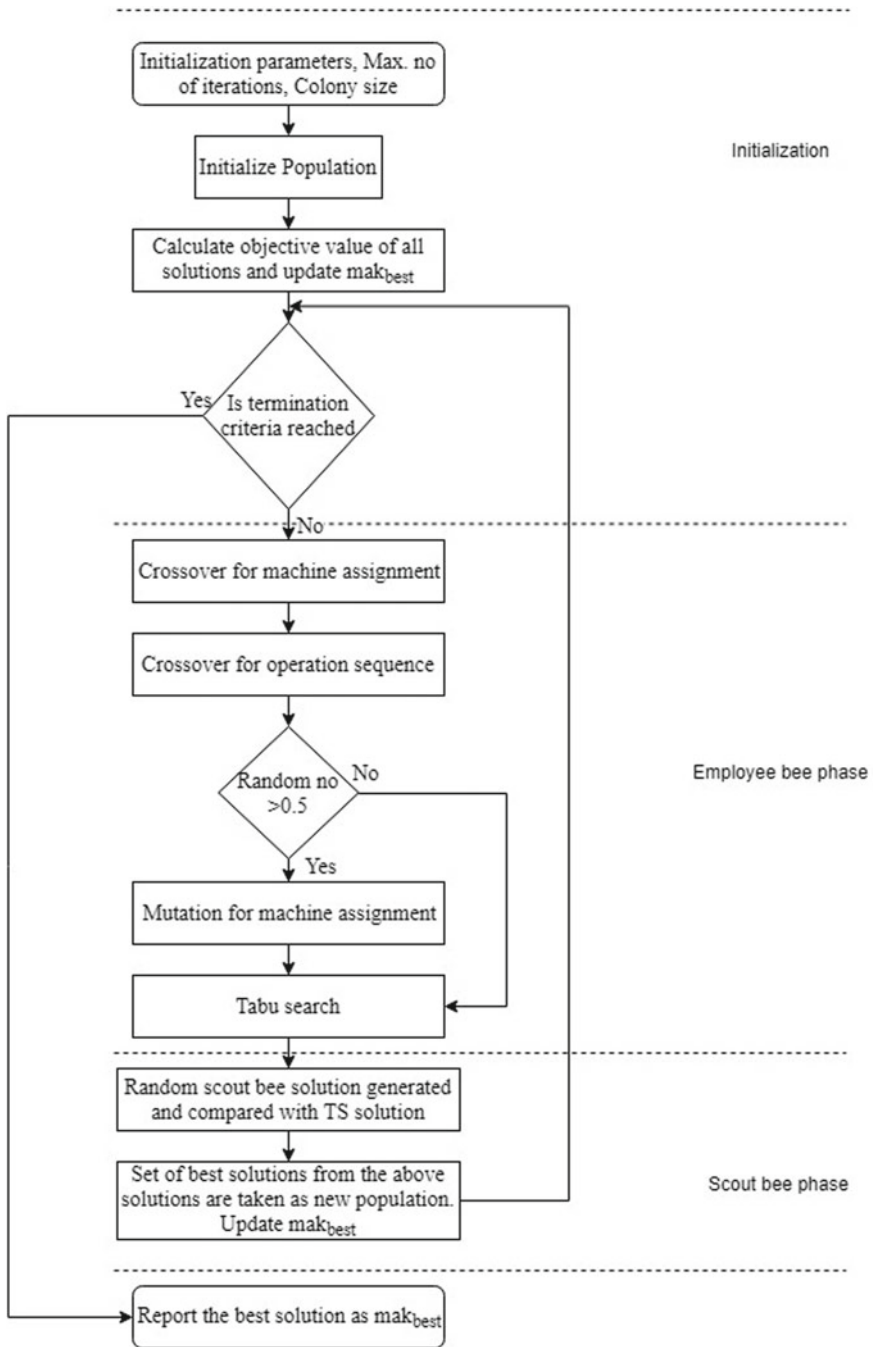


Fig. 4 Flowchart of the proposed hybrid ABC

Table 1 Results of benchmark problems

S. No.	Problem size	Xing's algorithm [14]	HTSA [17]	AIA [16]	BBO [18]	TLBO [15]	Proposed ABC
1	4 × 5	11	11	NA	11	11	11
2	8 × 8	14	14	14	14	14	14
3	10 × 7	11	11	NA	NA	11	11
4	10 × 10	7	7	7	7	7	7
5	15 × 10	11	11	11	12	12	11
6 (MK01)	10 × 6	42	40	40	40	40	40
7 (MK02)	10 × 6	28	26	26	28	28	27
8 (MK03)	15 × 8	204	204	204	204	204	204
9 (MK04)	15 × 8	63	61	60	64	63	60
10 (MK05)	15 × 4	181	172	173	173	172	172
11 (MK06)	10 × 15	60	65	63	66	65	63
12 (MK07)	20 × 5	148	140	140	144	144	140
13 (MK08)	20 × 10	523	523	523	523	523	523
14 (MK09)	20 × 10	308	310	312	310	311	307
15 (MK10)	20 × 15	212	214	214	230	214	214

Note The significance of the “Bold” is to indicate that the proposed problem solution is better than benchmark problem solutions

5 Conclusion and Future Scope

This paper considers the NP-hard FJSP with the makespan criteria, and the same is solved by a proposed hybrid algorithm where a TS is incorporated in the ABC algorithm. The effectiveness of using TS as a local search is highlighted by the improved exploration and convergence capability of the ABC algorithm in the local search space. The proposed hybrid ABC algorithm showed superior performance in comparison with other well-known meta-heuristics on Kacem's and Brandimarte's instances.

This work can be extended to solve other scheduling problems with constraints.

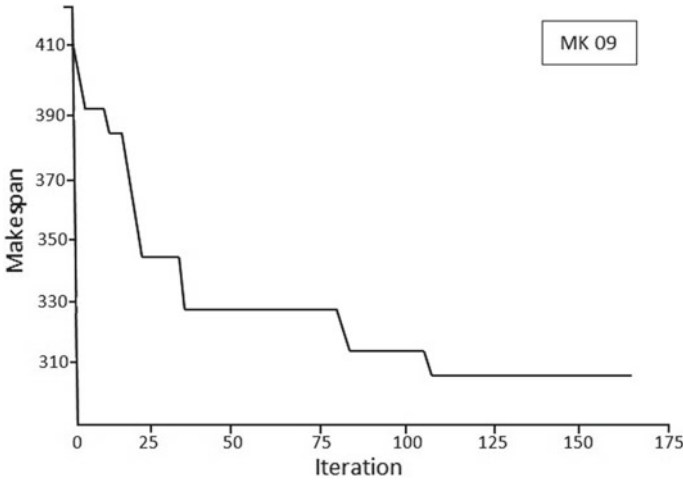
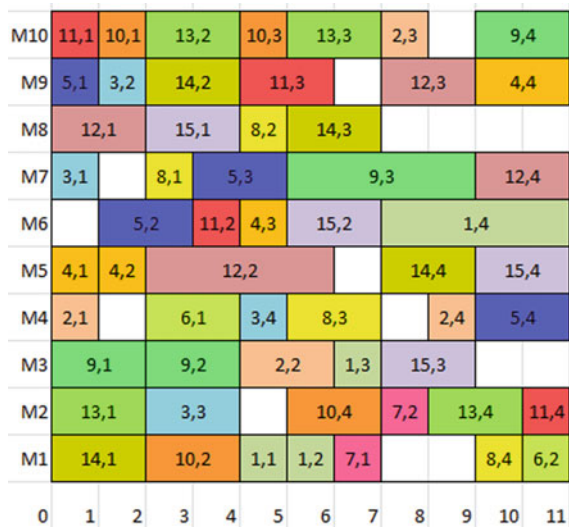


Fig. 5 Convergence of MK09 instance

Fig. 6 Gantt chart of Kacem’s 10 × 15 instance



References

1. Brandimarte P (1993) Routing and scheduling in a flexible job shop by tabu search. *Ann Oper Res* 41(3):157–183
2. Sharma N, Sharma H, Sharma A (2018) Beer froth artificial bee colony algorithm for job-shop scheduling problem. *Appl Soft Comput* 68:507–524
3. Xiang W et al (2018) A novel artificial bee colony algorithm based on the cosine similarity. *Comput Ind Eng* 115:54–68

4. Sharma H, Bansal JC, Arya KV, Yang XS (2016) Lévy flight artificial bee colony algorithm. *Int J Syst Sci* 47(11):2652–2670
5. Thammano A, Phu-ang A (2013) A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem. *Procedia Comput Sci* 20:96–101
6. Zhang R, Shiji S, Cheng W (2013) A hybrid artificial bee colony algorithm for the job shop scheduling problem. *Int J Prod Econ* 141(1):167–178
7. Wang L, Zhou G, Xu Y, Wang S, Liu M (2012) An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. *Int J Adv Manuf Technol* 60(1–4):303–315
8. Li JQ, Xie SX, Pan QK, Wang S (2011) A hybrid artificial bee colony algorithm for flexible job shop scheduling problems. *Int J Comput Commun Control* 6(2):286–296
9. Banharsakun A, Sirinaovakul B, Achalakul T (2012) Job shop scheduling with the best-so-far ABC. *Eng Appl Artif Intell* 25(3):583–593
10. Karaboga D, Gorkemli B (2012) A quick artificial bee colony-qABC-algorithm for optimization problems. In: *International symposium on innovations in intelligent systems and applications*. IEEE, pp 1–5
11. Kacem I, Hammadi S, Borne P (2002) Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 32(1):1–13
12. Saidi M, Fattahi P (2007) Flexible job shop scheduling with tabu search algorithms. *Int J Adv Manuf Technol* 32(5–6):563–570
13. Hurink J, Jurisch B, Thole M (1994) Tabu search for the job-shop scheduling problem with multi-purpose machines. *Oper Res Spektrum* 15(4):205–215
14. Xing LN, Chen YW, Yang KW (2009) An efficient search method for multi-objective flexible job shop scheduling problems. *J Intell Manuf* 20(3):283–293
15. Buddala R, Mahapatra SS (2018) An integrated approach for scheduling flexible job-shop using teaching–learning-based optimization method. *J Ind Eng Int* 1–12
16. Bagheri A, Zandieh M, Mahdavi I, Yazdani M (2010) An artificial immune algorithm for the flexible job-shop scheduling problem. *Futur Gener Comput Syst* 26(4):533–541
17. Li JQ, Pan QK, Liang YC (2010) An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Comput Ind Eng* 59(4):647–662
18. Rahmati SHA, Zandieh M (2012) A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem. *Int J Adv Manuf Technol* 58(9–12):1115–1129