International Journal of Advanced Engineering and Management Research
Vol. 2 Issue 5, 2017

http://ijaemr.com/ ISSN: 2456-3676

FLEXIBLE JOB-SHOP SCHEDULING BASED ON HYBRID ARTIFICIAL BEE COLONY ALGORITHM WITH DIVERSITY INDEX SEARCH FOR MULTIPLE DECISIONS MAKING IN MANUFACTURING SYSTEM

Li-Pen Chao[1], Tsun-Chen Lin[2] and [*]Chun-Liang Lu[3]

[1,3]Department of Applied Information and Multimedia, Ching Kuo Institute of Management and Health, Keelung County, Taiwan

[2]Department of Computer and Communication Engineering, Dahan Institute of Technology, Hualien County, Taiwan.

ABSTRACT

Optimization of flexible job-shop scheduling problem (FJSP) is of significant importance to the implementation of real-world manufacture. The fact to make the good decisions, respond appropriately and cost control of manufacturing system is essential for decision makers. However, selecting an action among a set of alternatives becomes harder when the decision making process involves several criteria. In this paper, an efficient scheme of combining Artificial Bee Colony algorithm with Diversity Index Search (ABC-DIS) is developed to simultaneously solving for the minimization of makespan, critical machine workload, and total workload on FJSP. The sequential operation-machine assignment (SOMA) of encoding representation is employed to always produce feasible candidate solutions in search space. Then the proposed ABC-DIS model by using a non-dominated sorting strategy is capable of solving for the multi-objective FJSP non-dominated solutions. Computational experiments are carried out using several benchmark dataset with various sizes and compared the ABC-DIS approach to other methods reported in some existing literature works. Experimental results show that the proposed approach is capable of achieving high quality, wide range of non-dominated solutions. In addition, the more diversity of the results in the same non-dominated solution as multiple decisions making are applicably providing for manufacturing system.

Key Words: Multiple decisions making, Flexible job-shop scheduling problem, Artificial bee colony algorithm, Diversity index search

1. INTRODUCTION

Decision making is the process of evaluation from available feasible alternatives. Appropriate selecting an action among a set of alternatives becomes harder when the decision making process involves several criteria rather than a single criterion. Further, most industrial practitioners are made under multiple and often conflicting criteria to satisfy a multitude of design criteria at the same time. In the fields of production management, the job-shop scheduling problem (JSP) is one of the most important issues requires lower cost and responds precisely to the market demands. The classical JSP consists of *n* jobs and *m* machines, which concerned with the allocation of resources for each job is processed on machines in a given order with a given processing time and each machine can process only one job at a time [1]. In contrast, the flexible job-shop scheduling problem (FJSP) [2] is an extension of the JSP where operations are allowed to be processed on a group of available machines satisfying some predefined constraints. It is more complex than JSP and well-known as an NP-hard problem.

Multi-objective FJSP (MOFJSP) has been studied by many researchers in recent years. Evolutionary algorithm (EA) is an effective type of meta-heuristic method, including Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4], Simulated Annealing (SA) [5], and other approaches including Artificial Bee Colony algorithm (ABC) [6], have acquired a lot of attention from researchers in this area. Due to its simple structure and outstanding performance, the ABC has received growing interest and has been widely used to solve many real-world optimization problems. Xia and Wu [4] applied the combination of PSO and simulated annealing algorithm (SA) to solve the problem. The PSO-SA adopted a weighted concept to convert multi-objectives into single-objective problem. Most of the literature used aggregated single-objective algorithms usually show lower level of performance as compared to Pareto-based multi-objective evolutionary algorithms (MOEAs). Ho and Tay [7] combined multi-objective evolutionary algorithm with guided local search (MOEA-GLS) to estimate bounds and obtained better results of non-dominated solutions than the PSO-SA method. The increasing need to seek a scientific support of multiple decisions making was not considered in practice manufacturing system. Lu et al. [8] later investigated non-dominated multi-objective PSO algorithm to search about diversity non-dominated solutions for FJSP. However, the diversity measurement under the same non-dominated solution is in the presence of uncertainty during evolutionary stage process.

In this paper, the proposed Diversity Index Search (DIS) strategy is merged into the artificial bee colony algorithm for non-dominated diversity search to tackle the problem mentioned above. The previously developed Segment Operation-Machine Assignment (SOMA) [8] encoding representation, which can always produce the feasible candidate solutions of FJSP, is employed to construct the ABC-DIS model for identifying the variety under the same non-dominated solutions. Furthermore, the more diversity in the same non-dominated set as to multiple decisions making are appropriately providing for manufacturing system. The rest of the paper is organized as follows. In section 2, the related works including FJSP and basic concept of ABC algorithm are described. The SOMA encoding representation, Diversity Index Search strategy and the proposed ABC-DIS framework are illustrated in section 3. Experiment results are provided in section 4. Finally, conclusions are made in section 5.

2. RELATED WORKS

2.1 Problem definition of FJSP

The scheduling problem in the FJSP [2] is divided into two sub-problems of routing and sequencing. In the routing phase, each operation will be assigned to the available machine. While in the sequencing stage, the operation position will depend on the order of the assigned operations for each machine. Some assumptions for FJSP are described as follows.

(1) A set of execution of *n* jobs $J_i (i = 1, 2, ..., n)$ and a group of available *m* machines $M_k (k = 1, 2, ..., m)$.

(2) Each job J_i needs O_j operations on the order of restraint using use *k*-th machine, $O_{i,j,k} = \{O_{i,1,k}, O_{i,2,k}, \cdots, O_{i,j,k}\}$.

(3) Each operation requires one machine to be executed from a set of available machines.

(4) All jobs and machines are available at time 0, and each machine can only execute one operation at a given time.

(5) The processing time of an operation on machine is predetermined, and the started operation cannot be interrupted.

Let C_i be the completion time of job J_i ,where $C_{i,j,k}$ means the required time of *j*-th operation of *i*-th job processed on *k*-th machine, where $1 \le i \le n,\ 1 \le j \le O_i,\ 1 \le k \le m$. W_k is the summation of processing time of operations that are processed on machine M_k. Three objectives, namely makespan (C_{max}), total workload (W_T), and critical workload (W_{CL}) are to be minimized simultaneously, where $C_{max} = \max\{C_i \mid i = 1, 2, \cdots, n\}$, $W_T = \sum_{k=1}^{m} W_k$, $W_{CL} = \max\{W_k \mid k = 1, 2, \cdots, m\}$, respectively.

2.2 Basic concept of Artificial Bee Colony (ABC) algorithm

Swarm intelligence ABC algorithm was first introduced in [9] and inspired from the foraging behavior of bee colony. In the basic ABC algorithm, there are three kinds of bees, namely employed bees, onlooker bees and scout bees. A bee that is currently exploiting a food source is called an employed bee, where each solution to the problem under consideration is called a food source. An onlooker bee is the one who waits on the dance area for making decision to choose a food source. A bee carrying out random search for a new food source is named a scout. The solutions of the optimization problem are represented as the position of the food sources, and the quality of the associated solution corresponds to the nectar amount of food source. The framework of ABC is described as follows.

The population of ABC algorithm consists of *D*-dimensional vector $X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,D}\}$, where $i = \{1, 2, \cdots, SN\}$ and *SN* is the number of food sources. At the beginning of the process, each solution X_i is generated by lower and upper bounds $X_{\min}$ and $X_{\max}$, where $x_{i,j} = x_{\min,j} + rand(0,1) \times (x_{\max,j} - x_{\min,j})$, $j = \{1, 2, \cdots, D\}$. After initialization, the employed bee randomly selected a food source position and modified in their memory as $v_{i,j} = x_{i,j} + \phi_{i,j} \times (x_{i,j} - x_{k,j})$ for each individual X_i and the candidate V_i, where

$k = \{1, 2, \cdots, SN\}$ and $\phi_{i,j} \in [-1,1]$. The employed bees performed to select the better one from the old individual X_i and the candidate V_i, and complete the sharing of position information with the onlooker bees on the dance area. A food source is selected by an onlooker bee depending on the probability value p_{i} according to the following expression [10], where fit_{i} is the *i*-th solution evaluated by its employed bee and proportional to the nectar amount of the food source in the position *i*. If the quality of individual

$$p_{\mathrm{i}} = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \tag{1}$$

solution cannot be improved beyond a previous iteration, then discard this individual one and the scout generates a new solution to replace X_{i}. The procedure of ABC algorithm can be summarized as below.

Step 1: Initialize *SN* points in the search space, individual position, iteration number, and evaluate the fitness value of population.

Step 2: The employed bee phase:

Generate the candidate solution V_i, and evaluate the fitness value of candidate solutions $f(V_i)$.

If the new produced solution V_i is better than X_i, it replaces the old one; if not, a counter variable of visits is increased.

Step 3: Select the predetermined top percentage of candidate solutions as marked solution.

Step 4: The onlookers phase:

The probability of a food source *i* being selected, is computed using the fitness value p_{i} described in Eq. (1).

Onlookers follow employed foragers to search new solutions and update marked solution.

Step 5: The scouts phase:

Scouts exploit new food source, and keep the best solution found in the search space.

Checks if food source value is greater than a maximum number of visits? If so, replace it by generating a new food source.

Step 6: If the iterative condition is met, stop and output the best solution achieved so far, otherwise, go to Step 2.

3. METHODS

Artificial bee colony algorithm has been shown to be very effective to solve global optimization problems. However, the encoding mechanism of the ABC should be extremely important when trying to find solutions to a problem in a search space. In the ABC procedure, each solution is represented by a two-vector-based solution representation, and a flexible decoding strategy is designed to solve for FJSP. The particle representation named Segment Operation-Machine Assignment (SOMA) [8] scheme is presented to always produce feasible candidate solutions, and the repair mechanism to maintain candidate feasibility is not required. In this section, we first describe the SOMA encoding mechanism, the flexible decoding strategy, and show the FJSP candidate Gantt chart. Then, we present the Diversity Index Search structure, and the heuristic operation. Finally, the proposed hybrid ABC-DIS model framework is detailed for solving the FJSP.

3.1 Segment Operation-Machine Assignment (SOMA) encoding representation

The effective particle encoding representation that we previously published and detailed in [8, 11], each dimension contains three components: integer part (machine selection), decimal part (priority order) and real-value number (operation number). In Fig.1 is the structure of SOMA representation on each dimension. This two-vector-based encoding is flexible enough for solving FJSP to satisfy the precedence constraints and operations in each job by using the real-value number in ABC algorithm.

Operation number

Machine selection	Prioity order

(Integer number) (Decimal number)

Figure 1: The structure of SOMA encoding representation.

Here the example of FJSP (3 jobs, 3 machines, and 6 operations) is considered and illustrated in Table 1. Three Jobs J_1, J_2 and J_3 need to be processed by at least one of three machines M_1, M_2 and M_3. Each job contains several operations such as Job 1 is split into $O_{1.1}$, $O_{1.2}$, $O_{1.3}$, Job 2 is split into $O_{2.1}$, $O_{2.2}$, and Job 3 just only consists of $O_{3.1}$. The time consuming for each machine to operate each job is predetermined, such as $O_{1.1}$ assigned to M_1 is 3 time units; $O_{1.2}$ assigned to M_2 is 5 time units, etc.

Table 1: The example of operation schedules for FJSP.

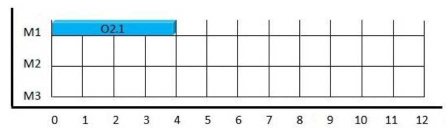
Machines \ Jobs	$O_{1.1}$	$O_{1.2}$	$O_{1.3}$	$O_{2.1}$	$O_{2.2}$	$O_{3.1}$
M_1	3	4	2	4	5	6
M_2	4	5	2	5	6	8
M_3	2	3	3	5	5	6

In order to explain how the SOMA scheme to produce feasible candidate solution, one randomly candidate encoding representation is shown as Figure 2. Then the detailed description of the flexible decoding strategy is illustrated in Step 1 to Step 6.

$O_{1.1}$	$O_{1.2}$	$O_{1.3}$	$O_{2.1}$	$O_{2.2}$	$O_{3.1}$
3.76	1.13	2.36	1.88	2.52	3.28

Figure 2: A possible candidate encoding representation of the individual.

Step 1: $O_{2.1}$ is assigned to M_{1}.



Candidate operation set O_1= { $O_{1.1}$, $O_{2.1}$, $O_{3.1}$ }.

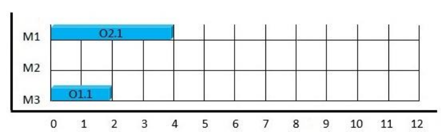
Candidate assignment set C_1= { $O_{1.1}$- M_3, $O_{2.1}$- M_1, $O_{3.1}$- M_3 }.

Priority order: Max($O_{1.1}$, $O_{2.1}$, $O_{3.1}$)=Max(0.76, 0.88, 0.28)=0.88.

Then the { $O_{2.1}$- M_1 } is assigned from C_{1}.

Figure 3-1: The candidate Gantt chart in Step 1.

Step 2: $O_{1.1}$ is assigned to M_{3}.



Candidate operation set O_2= { $O_{1.1}$, $O_{2.2}$, $O_{3.1}$ }.

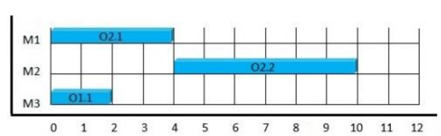
Candidate assignment set C_2= { $O_{1.1}$- M_3, $O_{2.2}$- M_2, $O_{3.1}$- M_3 }.

Priority order: Max($O_{1.1}$, $O_{2.2}$, $O_{3.1}$)=Max(0.76, 0.52, 0.28)=0.76.

Then the { $O_{1.1}$- M_3 } is assigned from C_{2}.

Figure 3-2: The candidate Gantt chart in Step 2.

Step 3: $O_{2.2}$ is assigned to M_{2}.



Candidate operation set O_3= { $O_{1.2}$, $O_{2.2}$, $O_{3.1}$ }.

Candidate assignment set C_3= { $O_{1.2}$- M_1, $O_{2.2}$- M_2, $O_{3.1}$- M_3 }.

Priority order: Max($O_{1.2}$, $O_{2.2}$, $O_{3.1}$)=Max(0.13, 0.52, 0.28)=0.52.

Then the { $O_{2.2}$- M_2 } is assigned from C_{3}.

Figure 3-3: The candidate Gantt chart in Step 3.

Step 4: $O_{3.1}$ is assigned to M_3.

Candidate operation set O_4= { $O_{1.2}$, $O_{3.1}$ }.

Candidate assignment set C_4= { $O_{1.2}$- M_1, $O_{3.1}$- M_3 }.

Priority order: Max($O_{1.2}$, $O_{3.1}$)=Max(0.13, 0.28)=0.28.

Then the { $O_{3.1}$- M_3 } is assigned from C_4.

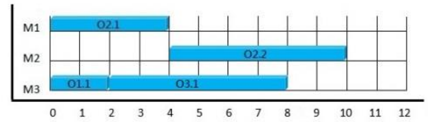


Figure 3-4: The candidate Gantt chart in Step 4.

Step 5: $O_{1.2}$ is assigned to M_1.

Candidate operation set O_5= { $O_{1.2}$ }.

Candidate assignment set C_5= { $O_{1.2}$- M_1 }.

Priority order: Max($O_{1.2}$)=Max(0.13)=0.13.

Then the { $O_{1.2}$- M_1 } is assigned from C_5.

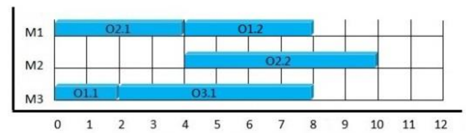


Figure 3-5: The candidate Gantt chart in Step 5.

Step 6: $O_{1.3}$ is assigned to M_2.

Candidate operation set O_6= { $O_{1.3}$ }.

Candidate assignment set C_6= { $O_{1.3}$- M_2 }.

Priority order: Max($O_{1.3}$)=Max(0.36)=0.36.

Then the { $O_{1.3}$- M_2 } is assigned from C_6.

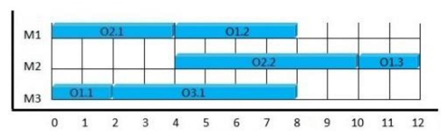


Figure 3-6: The candidate Gantt chart in Step 6.

3.2 Diversity Index Search (DIS) strategy

After the SOMA scheme to produce feasible candidate Gantt chart solutions, all job assignments are stored as a two dimensional matrix. Consider the example of FJSP in Table 1, the Diversity Matrix (DM) according to Gantt chart in Eq. (2) for each matrix element $d_{i,j}$, where $1 \le i \le n, 1 \le j \le t$, $n = 3, t = 12$. These operation numbers are labeled with relative time slots as $d_{i,j} = 1$, otherwise empty time slots are assigned as $d_{i,j} = 0$. In Eq. (3), the xor operation is employed to compare the two candidate Gantt chart solutions DM_i and DM_j. Based on the above xor operation analysis, $DM_i \otimes DM_j = 1$ indicates the two candidate Gantt charts are different, otherwise, $DM_i \otimes DM_j = 0$ denotes they are the same. During the DI index evolution procedure, it can utilize the information of the best individual found so far to enhance the solution diversity for FJSP.

$$DM = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,j} & \cdots & d_{1,t-1} & d_{1,t} \\ d_{2,1} & d_{2,1} & \cdots & d_{2,j} & \cdots & d_{2,t-1} & d_{2,t} \\ \vdots & \vdots & \ddots & \vdots & \iddots & \vdots & \vdots \\ d_{i,1} & d_{i,2} & \cdots & d_{i,j} & \cdots & d_{i,t-1} & d_{i,t} \\ \vdots & \vdots & \iddots & \vdots & \ddots & \vdots & \vdots \\ d_{n-1,1} & d_{n-1,2} & \cdots & d_{n-1,j} & \cdots & d_{n-1,t-1} & d_{n-1,t} \\ d_{n,1} & d_{n,2} & \cdots & d_{n,j} & \cdots & d_{n,t-1} & d_{n,t} \end{bmatrix}$$

(2)

$$DI = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N}(DM_i \otimes DM_j)}, \quad i \neq j$$

(3)

3.3 The proposed hybrid ABC-DIS model

Figure 4 shows the system architecture of our proposed hybrid ABC-DIS model. In this work, both the effective SOMA scheme that we previously published, and the DIS strategy are successfully merged into ABC algorithm for solving the multi-objective FJSP. Based on the SOMA encoding representation and DI index optimization approach mentioned above, detailed descriptions of the novel hybrid ABC-DIS procedure are illustrated as follows.

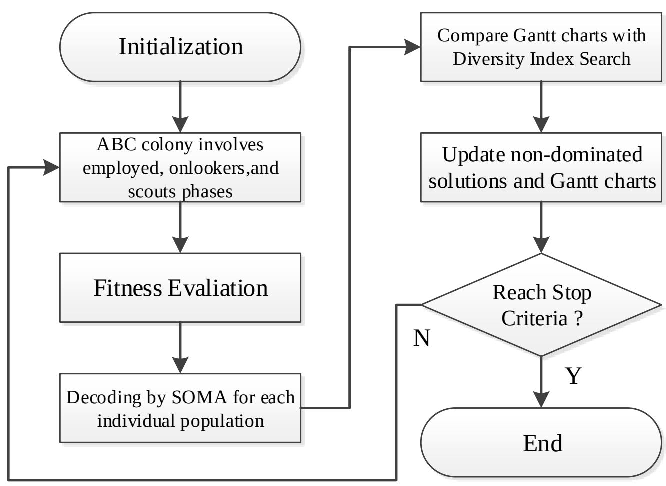


Figure 4: The system architecture of hybrid ABC-DIS model.

The ABC-DIS procedure: The proposed hybrid ABS-DIS model to solve for multi-objective FJSP.

Basic setting of parameters for the ABC algorithm.

Initialization randomly generates *SN* points in the search space.

Evaluate the fitness value for each individual population.

while Termination condition is not satisfied **do**

The employed bee phase:

    for *i*=1 to *SN* **do**

        Generate the candidate solution V_i .

        Evaluate the fitness value of candidate solution $f(V_i)$.

        Select the predetermined top percentage of candidate solutions as marked solution.

    end for

Calculate the probability p_i , set *t*=0, *i*=1.

The onlookers phase:

    while $t \le SN$ **do**

        if $rand(0,1) \le p_i$

            Generate the candidate solution V_i .

            Evaluate the fitness value of candidate solution $f(V_i)$.

            if $f(V_i) \le f(X_i)$ **then** $X_i = V_i$.

            set *t*=*t*+1

        end if

    end while

The scouts phase:

    Exploit new food source, and keep the best solution found in the search space.

Decoding by **SOMA** for each individual population.

Compare Gantt charts with **Diversity Index Search**.

Update multi-objective non-dominated solutions and Gantt charts.

26: **end while**

27: **Output** Global optimum solution(s) and FJSP diversity Gantt charts.

1. EXPERIMENT RESULTS

To illustrate the effectiveness and performance of the proposed hybrid ABC-DIS model, two popular representative benchmarks which are problem 10×10 and 10×7 with release date datasets [7] have been conducted to compare with other published methods, such as the PSO-SA [4], MOEA-GLS [7], VNGA [12] , and MOPSO-PDS [8] approaches. For each problem, the comparison results are reported in table contains three objectives: W_T (total workload), W_{CL} (critical workload), and C_{max} (makespan), which are mentioned in Section 2. The solutions found from these methods are shown in Table 2 to Table 3. The column labeled 'Gantt Chart' indicates diversity number and the symbol '$\times$' signifies that Gantt charts have not been provided. For giving an illustration, the two Gantt charts of each solution on problem 10×10 and 10×7 with release date datasets are exhibited in Figure 5.1 to Figure 6.6.

Table 2: ***Experimental Results for FJSP problem 10×10.***

Benchmark / Methods	W_T (total workload)	W_{CL} (critical workload)	C_{max} (make span)	Gantt charts diversity
PSO-SA [4]	44	6	7	1
MOEA-GLS [7]	41	7	8	$\times$
	42	5	8	$\times$
	43	5	7	$\times$
	42	6	7	$\times$
VNGA [12]	41	7	8	$\times$
	42	5	8	$\times$
	43	5	7	$\times$
	42	6	7	$\times$
MOPSO-PDS [8]	41	7	8	35
	42	5	8	16
	43	5	7	25

	42	6	7	13
Proposed ABC-DIS	41	7	8	66
	42	5	8	42
	43	5	7	58
	42	6	7	34

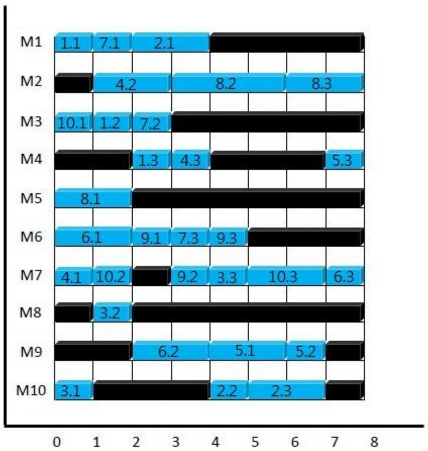


Figure 5.1: Gantt chart **I** for solution (41, 7, 8)

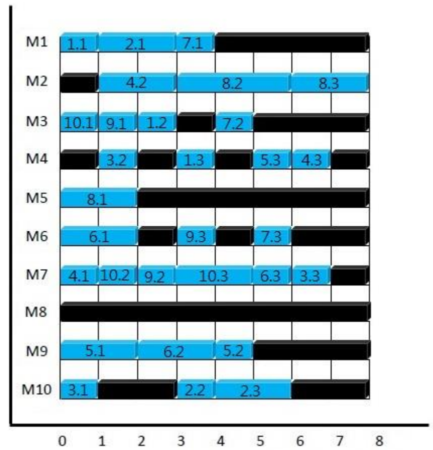


Figure 5.2: Gantt chart **II** for solution (41, 7 ,8)

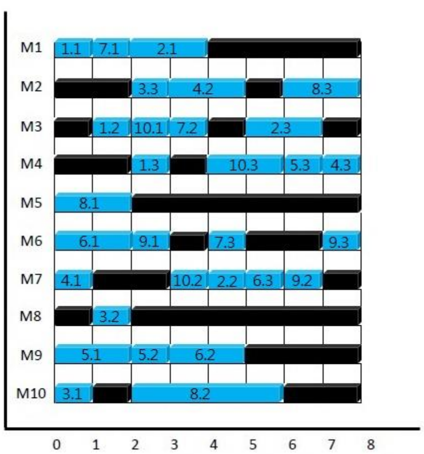


Figure 5.3: Gantt chart **I** for solution (42, 5, 8)

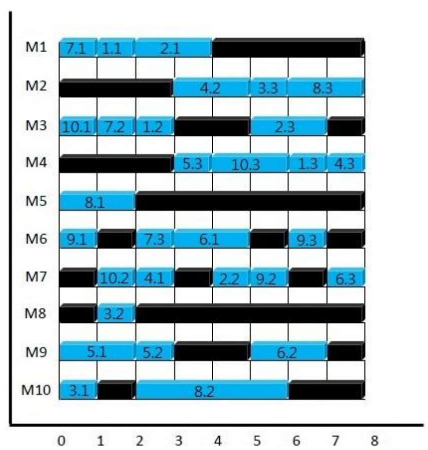


Figure 5.4: Gantt chart **II** for solution (42, 5, 8)

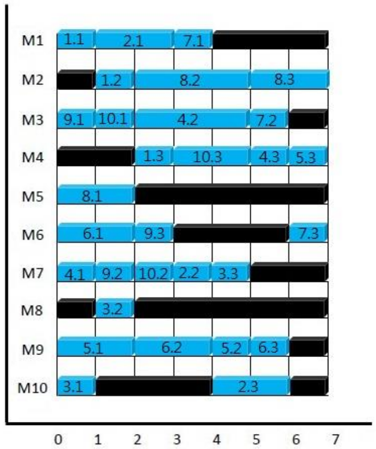


Figure 5.5: Gantt chart **I** for solution (42, 6, 7)

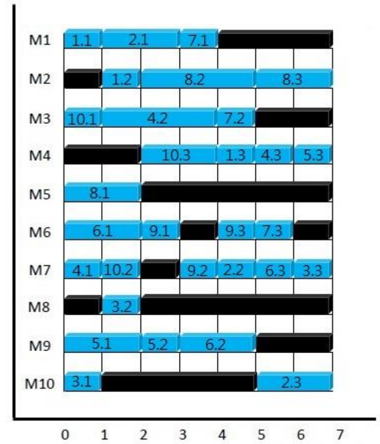


Figure 5.6: Gantt chart **II** for solution (42, 6, 7)

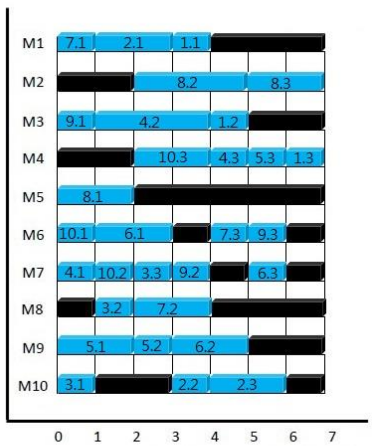


Figure 5.7: Gantt chart **I** for solution (43, 5, 7)

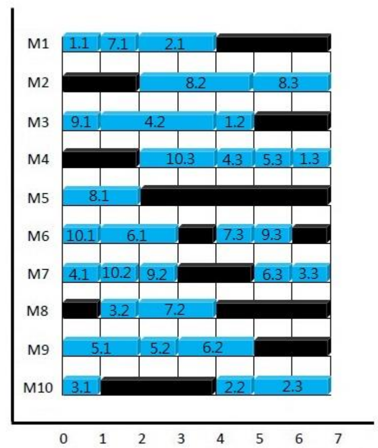


Figure 5.8: Gantt chart **II** for solution (43, 5, 7)

*Table 3**: Experimental Results for FJSP problem 10×7 with release date.***

Methods \ Benchmark	W_T (total workload)	W_{CL} (critical workload)	C_{max} (make span)	Gantt charts diversity
AL-CGA [7]	60	12	16	1
	61	11	15	1
	63	10	18	1
	64	10	17	1
	66	10	16	1

MOEA-GLS [7]	60	12	16	×
	61	11	15	×
	62	10	15	×
MOPSO-PDS [8]	60	12	16	3
	61	11	15	2
	62	10	15	4
Proposed ABC-DIS	60	12	16	22
	61	11	15	36
	62	10	15	16

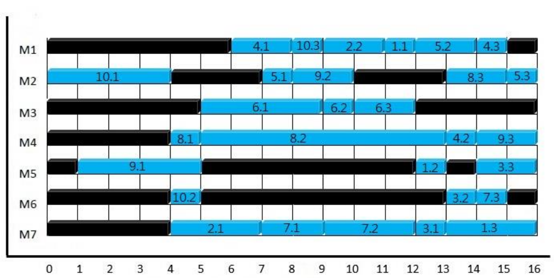


Figure 6.1: Gantt chart **I** for solution (60, 12, 16)

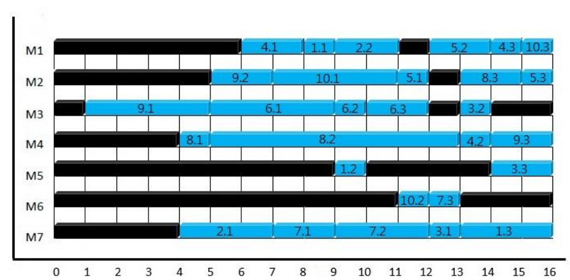


Figure 6.2: Gantt chart **II** for solution (60, 12, 16)

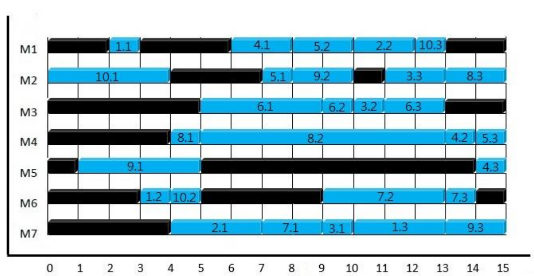


Figure 6.3: Gantt chart **I** for solution (61, 11, 15)

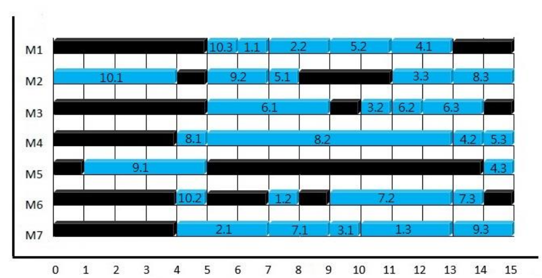


Figure 6.4: Gantt chart **II** for solution (61, 11, 15)

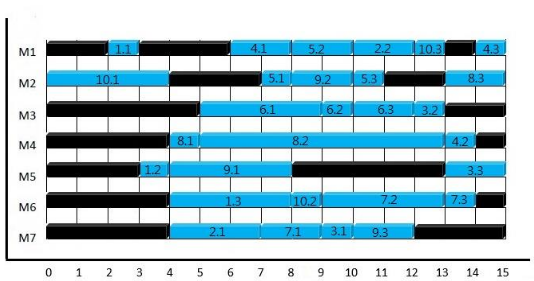


Figure 6.5: Gantt chart **I** for solution (62, 10, 15)

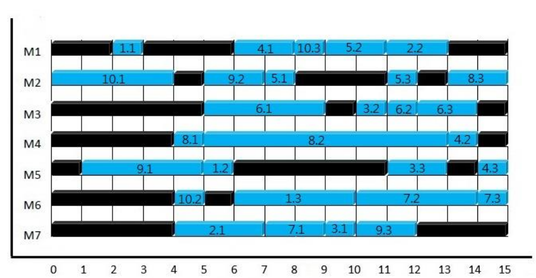


Figure 6.6: Gantt chart **II** for solution (62, 10, 15)

The comparison results show that the proposed algorithm offers the highest solution quality and outperforms better than PSO-SA for solving all of two benchmarks. Although the ABC-DIS finds the numbers of non-dominated solutions are equal to MOEA-GLS, VNGA, and MOPSO-PDS, the more Gantt chart diversity solutions can be found and has obvious superiority over other contenders.

5. Conclusions

In this paper, the individual encoding representation named Segment Operation-Machine Assignment (SOMA) is used to always produce feasible candidate solutions for the FJSP, a repair mechanism such as local search method to maintain candidate feasibility is not required. Furthermore, the developed solution searching strategy called Diversity Index Search (DIS) is adopted to enhance the ABC algorithm for finding different types of Gantt chart under the same non-dominated solutions. To validate the performance of the proposed hybrid ABC-DIS algorithm, two representative benchmarks in the literature is evaluated to compare with the other published PSO-SA, MOEA-GLS, VNGA, AL-CGA, and MOPSO-PDS methods. The experimental results on different-scale benchmarks show the effectiveness and better performance of the hybrid ABC-DIS model. In addition, the more diversity of Gantt charts can be found in multiple decisions making are applicably providing for manufacturing system.

References

[1] J. Kuhpfahl, C. Bierwirth, “A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective,” Computers & Operations Research, Vol. 66, pp. 44-57, 2016.

[2] Andy Ham, “Flexible job shop scheduling problem for parallel batch processing machine with compatible job families,” Applied Mathematical Modelling, Vol. 45, pp. 551-562, 2017.

[3] J. F. Gonçalves, J. J. M. Mendes, M. G.C. Resende, “A hybrid genetic algorithm for the job shop scheduling problem,” European Journal of Operational Research, Vol. 167, pp. 77-95, 2005.

[4] W. Xia and Z. Wu, “An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems,” Computers and Industrial Engineering, Vol. 48, pp. 409-425, 2005.

[5] F.D. Chou, H.M. Wang, P.C. Chang, “A simulated annealing approach with probability matrix for semiconductor dynamic scheduling problem,” Expert Systems with Applications, Vol. 35, pp. 1889-1898, 2008.

[6] R. Zhang, “An Artificial Bee Colony Algorithm Based on Problem Data Properties for Scheduling Job Shops,” Procedia Engineering, Vol. 23, pp. 131-136, 2011.

[7] N. B. Ho and J. C. Tay, “Solving Multiple-Objective Flexible Job Shop Problems by Evolution and Local Search,” IEEE Transactions on Systems, Man, and Cybernetics, Part C, Vol. 38, pp. 674-685, 2008.

[8] C.L. Lu, Shih-Yuan Chiu, Hsiang-Chun Cheng and Shi-Jim Yen, “Solving the Flexible Job-Shop Scheduling Problem Based on Multi-Objective PSO with Pareto Diversity Search,” International Journal of Intelligent Information Processing, Vol. 4, pp. 70-81, 2013.
[9] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-TR06, Erciyes University, 2005.
[10] P. Loubière, A. Jourdan, P. Siarry, R. Chelouah, “A sensitivity analysis method for driving the Artificial Bee Colony algorithm's search process,” Applied Soft Computing, Vol. 41, pp. 515-531, 2016.
[11] C.L. Lu, S.Y. Chiu, Chih-Hsu Hsu and Shi-Jim Yen, “Enhanced Differential Evolution Based on Adaptive Mutation and Wrapper Local Search Strategies for Global Optimization Problems,” Journal of Applied Research and Technology, Vol. 12, pp. 1131-1143, 2014.
[12] G. Zhang, L. Gao, and Y. Shi, “A Variable Neighborhood Genetic Algorithm for Multi-objective Flexible Job-Shop Scheduling Problems,” In Advanced Materials Research, pp. 369-373, 2010.