IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 24, 2020, accepted August 8, 2020, date of publication August 11, 2020, date of current version August 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015796

Metaheuristic for Solving Multi-Objective Job Shop Scheduling Problem in a Robotic Cell

XIAOHUI LI[1], (Member, IEEE), XI YANG[1], YI ZHAO[1], (Member, IEEE), YING TENG[2], AND YUAN DONG[1]

[1]School of Electronics and Control Engineering, Chang'an University, Xi'an 710054, China
[2]School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 610051, China

Corresponding author: Ying Teng (tengy@uestc.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Key Project of National Internet of Things Integrated Innovation and Integration under Grant 2018-470, in part by the Shaanxi Key Project of Research and Development Plan under Grant 2019ZDLGY03-01, and in part by the Xi'an Science and Technology Project under Grant 201805045YD23CG29.

ABSTRACT This paper deals with the multi-objective job shop scheduling problem in a robotic cell (MOJRCSP). All the jobs are processed according to their operations order on workstations. Different from classical job shop scheduling problem, the studied problem considers that jobs' transportation is handled by a robot. Also, the jobs are expected to be finished in a time window, instead of a constant due date. A mixed Integer Programming (MIP) model is proposed to formulate this problem. Due to the special characteristics of the studied problem and its NP-hard computational complexity, a metaheuristic based on Teaching Learning Based Optimization (TLBO) algorithm has been proposed. The proposed algorithm determines simultaneously the operations' assignments on workstations, the robot assignments for transportation operations, and the robot moving sequence. The objective is to minimize the makespan and the total earliness and tardiness. Computational results further validated the effectiveness and robustness of our proposed algorithm.

INDEX TERMS Robotic cell, job shop, multi-objective optimization, local search, teaching-learning based optimization.

I. INTRODUCTION

Intelligent production equipment has been gradually applied in recent manufacturing industry. As the core unit of intelligent workshop, the robotic cell (RC) is composed of unified information control system, material transportation robots and a set of processing equipments. The jobs can be transported between the workstation and the loading/unloading station by the robot. And the robotic cell can be connected with various production and storage equipment, and the whole unit is controlled by information control system. Due to the high efficiency and flexibility of robotic cell, it can be widely used in various modern advanced industrial production fields [1]–[4]. However, the existence of industrial robots increases the complexity of production environment. Therefore, in the process of job shop scheduling, the decision-maker should not only consider the processing arrangement of jobs, but also the task allocation and the movement of robot in the cell. Therefore, the traditional job shop scheduling can not be directly applied to this kind of advanced intelligent manufacturing production problem, so new production mode and new scheduling optimization scheme are needed to solve the scheduling problem of robotic cell. Thus, it becomes an important research topic to develop an effective collaborative optimal scheduling method (to find the optimal jobs processing plan and robot handling plan) to solve the robot cell scheduling problem. Yan *et al.* [5] deals with a real-time dynamic job-shop scheduling problem in a robotic cell. The studied problem can be modelled as a job shop where the jobs have to be transported between machines by robots. Elmi and Topaloglu [6] solved the cyclic job shop robotic cell scheduling problem with multiple robots. All the jobs are processed in the order of their operations on multiple machines with standard processing times and the single gripper robots. The robots perform the transportation operations of jobs between the stations.

There are different research methods to solve the scheduling problem of robotic cell. and common solutions mostly

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Zakarya.

include exact algorithms and approximate algorithms. The exact algorithm usually uses the method of mathematical modeling to solve the problem. Caumond *et al.* [7] provided the mathematical formulation and optimal solutions for the Flexible Manufacturing Systems Scheduling Problem (FMSSP) with one vehicle. This linear formulation is different from the previously works, it takes into account the maximum number of jobs allowed in the system, limited input/output buffer capacities, empty vehicle trips and no-move-ahead trips simultaneously. Che *et al.* [8] addressed the multi-robot 2-cyclic scheduling problem in a no-wait robotic cell where two parts enter and leave the cell during each cycle. In his work, multiple robots are responsible for transporting parts between machines. They developed a polynomial algorithm to find the minimum number of robots for all feasible cycle times. Nattafa *et al.* [9] studied the scheduling problems with jobs of different families on parallel machines, where not all machines are qualified (eligible) to process all jobs. To solve this problem, an integer linear programming model and a constraint programming model are proposed.

For scheduling problem, approximate algorithms are more employed. Zhang and Xing [10] addressed the distributed limited-buffer flowshop scheduling problem with makespan minimization criterion for the first time. They successively proposed two constructive heuristics for generating fast schedule and providing good initialization of meta-heuristics. Lei [11] presented a particle swarm optimization for multi-objective job shop scheduling problem. He aims to simultaneously minimize makespan and total tardiness of jobs. Wang *et al.* [12] investigated the energy-efficient distributed permutation flow shop scheduling problem (DPFSP) with the objectives of makespan and energy consumption. A multi-objective whale swarm algorithm (MOWSA) is proposed to solve this energy-efficient DPFSP. Also, to enhance energy efficiency without affecting production efficiency, a new problem-dependent local search is developed to improve the exploitation capability of MOWSA. Kurdi [13] proposes an improved island model memetic algorithm with a new naturally inspired cooperation phase (IIMMA) for multi-objective job shop scheduling problem. The new cooperation phase is mainly used to improve the exploitation capabilities of algorithm. Udomsakdigool and Khachitvichyanukul [14] present ant colony algorithm for solving the multi-objective Job Shop Scheduling Problem. The objectives considered in this study include the minimization of makespan, mean flow time, and mean tardiness.

In order to accelerate the convergence of the algorithm, many works combined the traditional approximation algorithm with neighborhood search technology which can be more efficient in solving FMS scheduling problem. To prevent premature convergence in a multi-objective optimization problem, the SA (Simulated annealing) algorithm is utilized as local search in the works of Mokhtari and Hasani [15]. It is capable of obtaining more solutions by escaping from the local minima in a wider solution space. Arnold and Sorensen [16] combined three powerful local search techniques, and implemented them in an efficient way. In their work, experiments have been made to determine how local search can be effectively combined with perturbation and pruning and how to guide the search to better solutions. Moghaddam *et al.* [17] proposed a new multi-objective Pareto archive particle swarm optimization (PSO) algorithm combined with genetic operators as variable neighborhood search (VNS). Their methods provided a better solution for large-sized problem instances within a reasonable computational time. Moslehi and Mahnam [18] present a new approach based on a hybridization of the particle swarm and local search algorithm to solve the multi-objective flexible job-shop scheduling problem. Zhang *et al.* [19] has formulated the textile dyeing process scheduling problem as a bi-objective optimization model, in which one objective is related with tardiness cost while the other objective reflects the level of pollutant emission. And they proposed a multi-objective particle swarm optimization algorithm enhanced by problem-specific local search techniques (MO-PSO-L) to seek high-quality non-dominated solutions. Luo *et al.* proposed a Distributed Flexible Job Shop Scheduling Problem with Transfers (DFJSPT), in which operations of a job can be processed in different factories. In order to expand the search space and accelerate the convergence speed of the solution, an Efficient Memetic Algorithm (EMA) is proposed to solve the DFJSPT with the objectives of minimizing the makespan, maximum workload, and total energy consumption of factories.

As a matter of fact, previous works mainly focus on the cases of single objective scheduling problem with robots or multi-objective scheduling problem without transported robots. These proposed solutions cannot address well our studied problem (MOJRCSP). Furthermore, most of the research methods are based on exact algorithms, and few approximate algorithms are available, especially for multi-objective manufacturing workshops with robotic cells. Therefore, in this work, we are aiming at solving the MOJRCSP with improved multi-objective teacher learning-based optimization algorithm (IMOTLBO). Since TLBO algorithm needs fewer parameters, it is simple and has strong convergence ability and fast convergence speed, so it has been applied in various fields [21]–[24], [26]–[28]. For multi-objective teacher learning-based optimization algorithm (MOTLBO) problem, up to now, the application of MOTLBO has been very limited. In [29], a modified version of the TLBO algorithm is introduced and applied for the multi-objective optimization of heat exchangers by Rao *et al.* Plate-fin heat exchanger and shell and tube heat exchanger are considered for the optimization. Maximization of heat exchanger effectiveness and minimization of total cost of the exchanger are considered as the objective functions. Toğan [22] presents a procedure employing a Teaching-Learning Based Optimization (TLBO) technique for discrete optimization of planar steel frames. The design algorithm aims to obtain minimum weight frames subjected to strength and displacement requirements imposed. In this work, our optimization goal is the minimization of makespan and the total weighted earliness and tardiness. In order to accelerate the convergence of the algorithm, we combined the TLBO algorithm with a powerful local search technique: variable neighborhood descent (VND). The algorithm determines the optimal sequence of jobs operation assignment and the robot movement in scheduling. Finally, with different instances the proposed IMOTLBO is compared with multi-objective algorithms, *i.e.*, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [23], the Pareto-based Grouping Discrete Harmony search algorithm (PGDHS) [24] and the Multi-Objective Particle Swarm Optimization (MOPSO) [25]. The results proved the proposed algorithm's effectiveness and advantage.

Contribution and innovation of this study are highlighted as follows:

- The multi-objective job shop robotic cell scheduling problem is formulated as a Mix Integer Programming model with transportation by handling robot, jobs' finished time windows are considered.
- This study proposes a new hybrid multi-objective metaheuristic which combines the TLBO algorithm with a powerful local search technique, the experimental results show that our algorithm obtains better non-dominated solutions than others.

The rest of this paper is organized as follows: The problem description and mathematical model is given in Section 2. In Section 3, the details of our proposed IMOTLBO is presented for solving the MOJRCSP. Experimental results and the conclusion are given in Section 4.

II. PROBLEM DEFINITION AND MATHEMATICAL FORMULATION

A. PROBLEM DESCRIPTION

A typical robotic cell consists of M workstations, a load/unload station and a handling robot. The framework of the RC system is illustrated in Figure 1. A number of jobs $J = \{J_1, \ldots J_N\}$ arrive at the load station of RC system and each of them has its own processing sequences. Each job consists of a number of n_j operations $O_{j1}, \ldots O_{jn_j}$ specifying the job's visiting route and different processing times p_{ji}. The number of operations of each job n_j may be different and $n_j \leq M$.

Without loss of generality, each job starts from the load station and finishes on the unload station. An automated handling robot moves jobs from one station to another. A move task Tr_{ji} indicates that the robot transports job j from station $\mu_{j,i-1}$ to station μ_{ji}, where μ_{ji} corresponds to the processed station of operation O_{ji}. Before move Tr_{ji}, if the robot is not in the position $\mu_{j,i-1}$, an empty move Tr'_{ji} is needed to arrive the desired position. The robot's loaded transportation and empty move should be considered in the scheduling problem. Robot's loading time ε_c and unloading time ε_d are set to a constant value.

Each operation O_{ji} starts the process on station μ_{ji} at its arriving time s_{ji} and leave the station at time t_{ji}, its staying

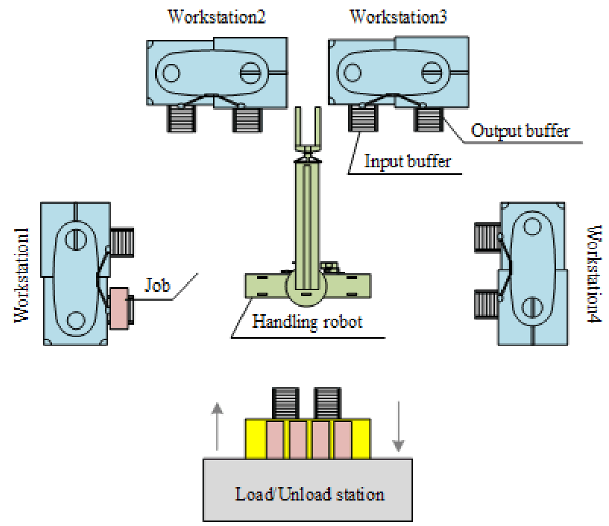

FIGURE 1. A typical robotic cell with a material handing robot.

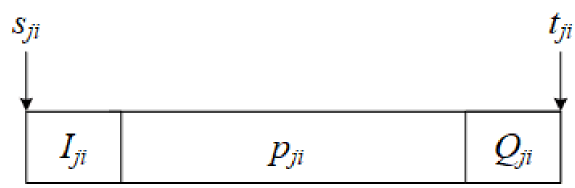

FIGURE 2. Variables denoted to the description of an operation.

time can be divided into three parts, p_{ji} represents the processing time of O_{ji}, I_{ji} and Q_{ji} correspond to the waiting time of O_{ji} in the input buffer and output buffer, respectively. These variables model the operation process on the station (shown in Figure 2).

The goal is to minimize the makespan and the total tardiness and earliness simultaneously, since each job's expected finish time is confined into time windows with the given minimal and maximal time bounds.

The following assumptions are considered:

- all jobs enter and leave the RC system through the load and unload stations;
- the capacity of load/unload station and input/output buffer of workstation is unlimited;
- workstation and robot can process at most one job at a time;
- each operation can only be processed on only one workstation;
- failures of machines and robot are ignored;
- at time zero, all jobs are arrived at the loading station, and all workstations and robot are available;
- no preemption is allowed.

B. NOTATIONS

1) NOTATIONS FOR PARAMETERS

J number of jobs, which are index as job j, $j \in [1, J]$

M number of stations, $1, \ldots M$ indicates machines, 0 and $M+1$ present the load and unload stations

n_j number of operations of job j, 0 and n_j+1 present the operation at the load and unload stations
O_{ji} the i^th operation of job j
p_{ji} processing time of O_{ji}
a_j lower bound of job j's due date
b_j upper bound of job j's due date
τ_{rl} loaded trip duration from station r to station l
σ_{rl} empty trip duration from station r to station l
ε_c loading time for one job
ε_d unloading time for one job
μ_{ji} station where O_{ji} is processed
Tr_{ji} Loaded transportation task between $O_{j,i-1}$ and $O_{j,i}$ (from $\mu_{j,i-1}$ to μ_{ji})
Tr'_{ji} empty moving task before Tr_{ji}
H a large positive number

2) NOTATIONS FOR VARIABLES

t_{ji} completion time of operation O_{ji}
Q_{ji} waiting time of operation O_{ji} in the output buffer of station μ_{ji}
I_{ji} waiting time of operation O_{ji} in the input buffer of station μ_{ji}
s_{ji} arriving time of O_{ji} at station μ_{ji}
C_j completion time of job j
E_j earliness of job j
T_j tardiness of job j
U_{jik} $=1$, if O_{ji} is processed before O_{jk}
$=0$, otherwise
$X_{jij'i'}$ $=1$, if O_{ji} is processed before $O_{j'i'}$ at station μ_{ji}
$=0$, otherwise
$Z_{jij'i'}$ $=1$, if Tr_{ji} is processed before $Tr_{j'i'}$
$=0$, otherwise

C. OBJECTIVE FUNCTION AND CONSTRAINTS

The objective function is:

$$f_1 = min(C_{max}) \tag{1}$$
$$f_2 = min(\sum_{j\in[1,J]} (E_j + T_j)) \tag{2}$$

The following constrains are considered:

$$t_{ji} = s_{ji} + I_{ji} + p_{ji} + Q_{ji} \tag{3}$$

where $j \in [1, J]$, $i \in [1, n_j]$.

This constraint ensures that the completion time of operation O_{ji} is equal to the arriving time plus its staying time at the station, where the staying time consists the waiting time in input buffer I_{ji}, the processing time p_{ji} and the waiting time in output buffer Q_{ji}.

$$t_{ji} = t_{j,i-1} + \tau(\mu_{j,i-1}, \mu_{ji}) + \varepsilon_c + \varepsilon_d + I_{ji} + p_{ji} + Q_{ji} \tag{4}$$

where $j \in [1, J]$, $i \in [1, n_j]$.

$$t_{j1} \geq \tau(0, \mu_{j1}) + \varepsilon_c + \varepsilon_d + I_{j1} + p_{j1} + Q_{j1} \tag{5}$$
$$t_{j,n_j+1} = t_{j,n_j} + \tau(\mu_{j,n_j}, M+1) + \varepsilon_c + \varepsilon_d \tag{6}$$
$$C_j = t_{j,n_j+1} \tag{7}$$

where $j \in [1, J]$.

These constraints ensures that the operations are performed according to the job's sequence, where $O_{j,i-1}$ is denoted as the immediate previous operation of O_{ji} in the job sequence. Constraint (4) computes the completion time t_{ji} of O_{ji}, it equals to the sum of completion time of the immediate previous operation $O_{j,i-1}$, the transportation time $\tau(\mu_{j,i-1}, \mu_{ji})$, the loading time ε_c and the unloading time ε_c, the staying time of O_{ji}.

Constraint (5) and (6) can be considered as the special case of (4). Constraint (5) computes the completion time of the first operation of each job with $t_{j0} = 0$. Constraint (6) and (7) calculates the completion time of each job, which is equal to the time when the job is entered to the unload station, where $I_{j,n_j+1} = p_{j,n_j+1} = Q_{j,n_j+1} = 0$.

$$t_{ji} + \tau(\mu_{ji}, \mu_{jk}) + \varepsilon_d + \varepsilon_c \leq s_{jk} + H(1 - U_{jik}) \tag{8}$$

where $j \in [1, J]$, $i, k \in [1, n_j]$.

This constraint ensures that two operations of the same job cannot be performed at the same time. If O_{ji} is processed before O_{jk} ($U_{jik} = 1$), operation O_{jk} must start after the robot moves the job from station μ_{ji} to the input buffer of μ_{jk}.

$$t_{ji} - p_{ji} - Q_{ji} \geq t_{j',i'} - Q_{j'i'} - H(1 - X_{jij'i'}) \tag{9}$$
$$X_{jij'i'} + X_{j'i'ji} \leq 1 \tag{10}$$

where $j, j' \in [1, J]$, $i \in [1, n_j]$, $i' \in [1, n_{j'}]$, with $\mu_{ji} = \mu_{j'i'}$

These constraints ensures that two operations on the same station cannot be processed at a time. If O_{ji} is processed before $O_{j'i'}$ on station μ_{ji} ($X_{jij'i'} = 1$), then $O_{j'i'}$ will start its process after O_{ji} enter the output buffer.

$$t_{ji} + \varepsilon_c + \tau(\mu_{ji}, \mu_{j',i'-1}) + \varepsilon_d + \sigma(\mu_{j',i'-1}, \mu_{j'i'}) \leq t_{j'i'} + H(1 - Z_{jij'i'}) \tag{11}$$
$$Z_{jij'i'} + Z_{j'i'ji} \leq 1 \tag{12}$$

where $j, j' \in [1, J]$, $i \in [1, n_j]$, $i' \in [1, n_{j'}]$

These constraints ensure that two loaded moves cannot be performed at the same time by the robot. In the case of $Z_{jij'i'} = 1$, the loaded move $Tr_{j'i'}$ must start after the previous loaded move Tr_{ji} is completed. When the robot finish the previous move, and if its current position μ_{ji} is different to the starting position $\mu_{j',i'-1}$ of loaded move Tr_{ji}, an empty move is required.

$$t_{ji} \leq t_{j'i'} + H(1 - X_{jij'i'}) \tag{13}$$
$$t_{ji} - Q_{ji} \leq t_{j'i'} - Q_{j'i'} + H(1 - X_{jij'i'}) \tag{14}$$
$$t_{ji} - Q_{ji} - p_{ji} \leq t_{j'i'} - Q_{j'i'} - p_{j'i'} + H(1 - X_{jij'i'}) \tag{15}$$
$$t_{ji} - Q_{ji} - p_{ji} - I_{ji} \leq t_{j'i'} - Q_{j'i'} - p_{j'i'} - I_{j'i'} + H(1 - X_{jij'i'}) \tag{16}$$

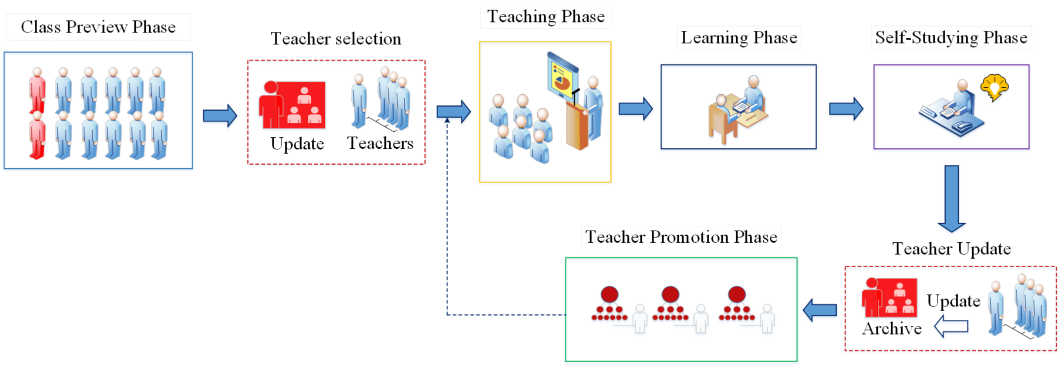

FIGURE 3. **Framework of IMOTLBO algorithm.**

where $j, j' \in [1, J]$, $i \in [1, n_j]$, $i' \in [1, n_{j'}]$, with $\mu_{ji} = \mu_{j'i'}$

These constraints ensure that the FIFO (First In First Out) buffer management rule is applied in the studied RC system, then all the successive operations are processed on the same order on the station, including input buffer, machine and output buffer.

$$C_{max} = Max(C_j) \tag{17}$$

$$E_j = Max(0, a_j - C_j) \tag{18}$$

$$T_j = Max(0, C_j - b_j) \tag{19}$$

where $j \in [1, J]$

Constraints (16), (17) and (18) compute the makespan, the earliness and tardiness of each job, respectively.

III. PROPOSED METAHEURISTIC

Teaching learning-based optimization (TLBO) is a new intelligent optimization algorithm proposed by Rao *et al.* [29]. The TLBO algorithm is a kind of population based metaheuristic which simulates the teaching learning process. In standard TLBO, there are two vital stages which are teaching phase and learning phase. The algorithm can be simply described as that the learners learn from the teacher (the best learner of the current population in the teaching phase). While they learn from other learners in the learning phase.

In this work, an Improved Multi-Objective Teaching Learning Based Optimization algorithm is proposed to solve the studied problem, where class preview phase, self-studying phase and teacher promotion phase are incorporated. The framework of IMOTLBO algorithm is shown in Figure 3, and the details are described as follows.

A. ENCODING SCHEME

Encoding scheme is an essential element of any metaheuristic algorithm. In this study, a solution to the problem (student or teacher) is represented by a ($len \times 2$) matrix where len is the sum of total number of operations and number of jobs (each job has a pseudo operation for moving the job to unload station). The first line represents the job's index, the

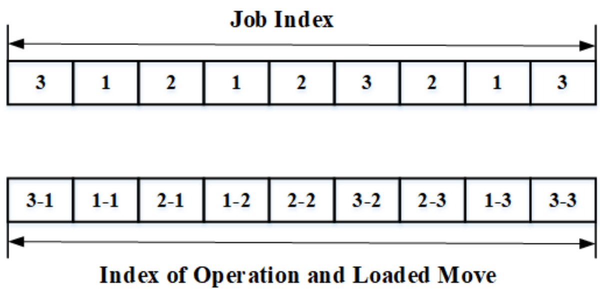

FIGURE 4. **Individual representation.**

number of occurrences of job j is equal to $n_j + 1$. The second line represents the scheduling sequence where each element indicates the index of operation O_{ji} and loaded move Tr_{ji}.

An example for the studied problem with 3 jobs is shown in Figure 4. In this example, one solution is represented as [3 1 2 1 2 3 2 1 3] in the first line where each job has 2 processing operations. In the relative position of the second line, the first number '3-1' represents the operation O_{31} and a loaded move Tr_{31} which transports job 3 from the load station to its first processing station. The second number '1-1' represents the operation O_{11} and the loaded move Tr_{11}, since the handling robot stopped on M_2 in the preview loaded move Tr_{31}, an empty move Tr'_{11} is required before Tr_{11} in order to move the robot to the starting station of Tr_{11}. The number '3-3' is a pseudo operation which indicate that the job 3 returns to the unload station. The information in the second line not only represents the scheduling sequence of operations, but also indicates the transportation sequence of handling robot. A gantt chart of the presented example is shown in Figure 5.

B. CLASS PREVIEW PHASE–POPULATION INITIALIZATION

In the teaching process, if the students have a good knowledge level before the teaching activities, it is beneficial to the teaching quality of the class. This part is defined as the class preview phase, which is responsible for generating the initial population.

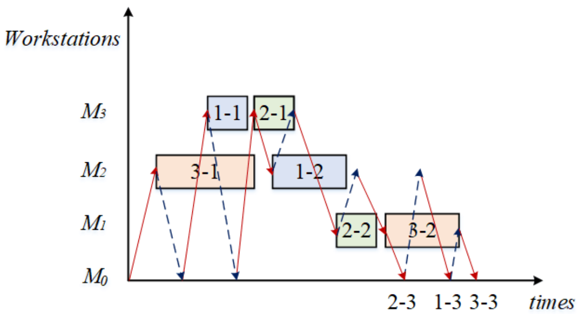

FIGURE 5. Gantt chart of example.

In this study, a well-known heuristic NEH (Nawaz, Enscore and Ham) [30] and dispatching priority rule SPT (Shortest Processing Time) [31] are applied to generate two initial solutions, in order to accelerate the convergence of the algorithm. The other solutions in the initial population are generated randomly.

C. TEACHER SELECTION–PARETO RANKING

Since our studied problem is a multi-objective optimization, more than one teachers are selected from the class according to personal knowledge level and performance. In IMOTLBO algorithm, the personal knowledge level and performance refer to objective values and ranking level of the solution. Each teacher has his own advantages, no one can completely surpass another. This process is based on Pareto dominance relationship. A same Pareto sort method of NSGA-II is applied, all solutions are divided into several Pareto fronts, the first Pareto front is considered as the teachers and they are copied into an archive.

D. TEACHING PHASE AND LEARNING PHASE–CROSSOVER

In the preview step, the best individuals are identified as the teachers. The teachers try to improve the performance of students through the communication of knowledge. The students obtain knowledge from the teachers using the crossover operator. The PTL technique [32] is applied in this section, it can produce a pair of different permutations, even from two identical individuals (see Figure 6). The new generated solutions are compared with the original one, the best one (who dominates the others) will be accepted. If there is no dominant relationship between them, then a random chosen solution is regarded as the student.

Since the processing sequence of operations in a job cannot be changed, the operations order should be updated by a repair process after the modification. Generally, as long as the solution is modified, the sequence of operations of jobs should be rechecked by repair process to ensure the feasibility of the solution.

In the learning phase, students improve their performance by the communication among themselves. A student X interacts with another student y which is chosen randomly. This

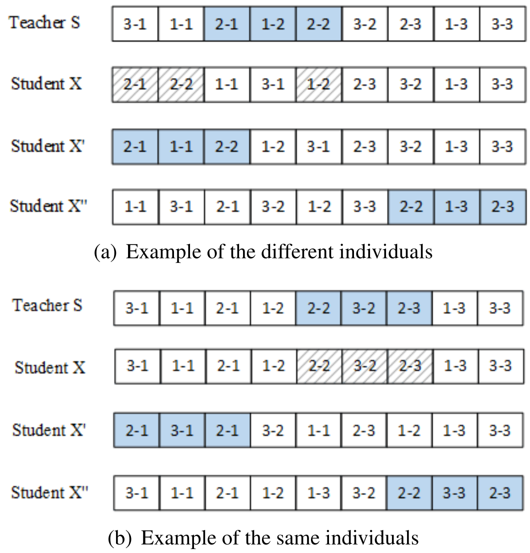

FIGURE 6. PTL crossover technique.

crossover is performed between students in the same manner as shown in Figure 6.

E. SELF-STUDYING PHASE–MUTATION

After the knowledge communication of the class, each student needs to review their knowledge by self study. In this phase, students complete self-improvement by a mutation operator. Each time, one of the neighborhood structures mentioned below (see Figure 7) is selected randomly as the mutation operator.

F. TEACHER PROMOTION PHASE–LOCAL SEARCH

In teaching activities, teachers continue to improve their knowledge and ability, which is beneficial to the performance of the whole class. The aim is to further improve the current optimal solutions of the entire population and archive. In this phase, a local search algorithm is applied to improve the individuals in the archive at the end of each iteration.

Usually, local search method receives one solution as input and returns one solution as output. In this work, all individuals in archive are considered as the input of local search and return a set of non-dominated solutions. This is done to keep the diversity of algorithm. IMOTLBO implements a local search technique based on the variable neighborhood descent (VND) algorithm [33], it moves continuously one solution to another position in the search space. In our implementation, we use $l_{max} = 4$ different neighborhood structures (see Figure 7), these neighborhood structures are briefly explained as follows:

- Exchange: This neighborhood structure exchanges the position of two operations.
- Reverse: This structure reverse the order of operations between two randomly selected position.
- Single point insert: This structure removes an operation from its current position and relocates it at a random position.

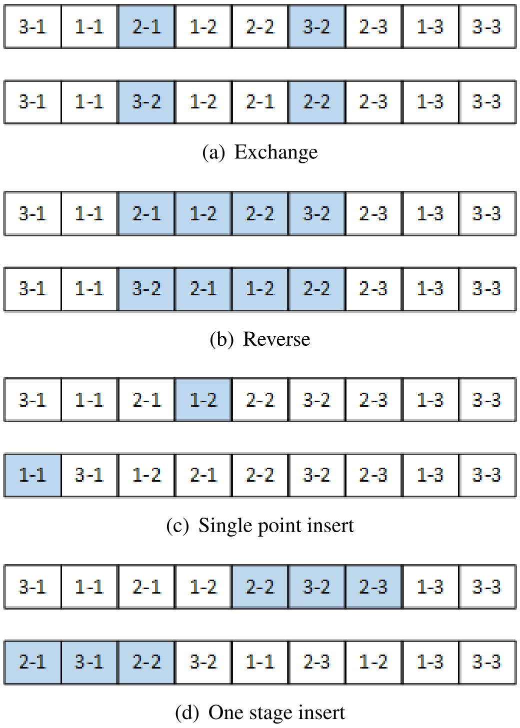

FIGURE 7. Example of neighborhood structures.

- One Stage Insert: In this structure, a part of operations between two random positions are removed to another position.

Inspired by the local search algorithm for E/T criterion of Gao *et al.* [24], an earliness and tardiness swap method is proposed to improve the individual. According to the completion time and expected due date interval of jobs, the jobs are divided into three groups in this method. These three group include the early finished jobs, the late finished jobs and on-time finished jobs, respectively. In this method, two adjacent operations on same are considered as pair from the first two operations. For all operations on all stations, if the left operation in the pair belongs to the job in group one, and the right operation in the pair belongs to the job in group two, then the position of the pair of operations on station will be interchanged. For example, as shown in Figure 5, suppose that the completion time of job 2 is before its earliest due date a_2, the completion time of job 3 is after its latest due date b_3, and the completion time of job 1 is between a_1 and b_1. The operation O_{22} and O_{32} will swap their positions.

The pseudo code of proposed local search technique is shown in Algorithm 1. The local search receives the individuals in archive as the input. The first step is to initialize the auxiliary populations *pop* and *localpop*. Then, the local search process is applied for each individual in *Arch*. Different from the standard local search technique, our improved local search is used to generate several non-dominated solutions. At the beginning of each iteration, the i_{th} solution in *Arch* is selected and considered as s, the temporary population *localpop* is initialized with the solution s. We set l to 1 and

Algorithm 1 The Pseudo-Code of Local Search Method

```
Input: The individuals in archive Arch
Output: The offspring Archive Arch
Begin
    pop ← φ
    localpop ← φ
    for i = 1 to nArch
        s ← Arch(i)
        localpop ← s
        l ← 1
        while l < l_max do
            s' ← implement l_th neighborhood move(s)
            if s is dominated by s' then
                s ← s', f(s) ← f(s')
                update(localpop, s')
                l ← 1
            elseif s and s' are non-dominated solutions do
                update(localpop, s')
                l ← 1
            else
                l ← l + 1
            end
        end while
        s' ← EarlineesTardinessSwap(s)
        update(localpop, s')
        update(pop, localpop)
        localpop ← φ
    end
    Arch ← pop
    return Arch
End
```

start the following search process. When a new solution s' is generated by implementing the neighborhood structures. By comparing s' and original solution s, there are three possibilities: 1) s' dominates s, then s is replaced by s' and *localpop* is updated, l is reset to 1 for restarting the searching process with the first neighborhood structure. 2) there are no dominance relationship between s and s', then, *localpop* is updated and l is reset to 1. 3) s' is dominated by s, then we set $l = l+1$ to implement the next neighborhood structure. When the above mentioned search is finished, Earliness-Tardiness Swap is applied and update *localpop* with the generated solution s'. At the end of each iteration, the auxiliary population *pop* is updated by *localpop*, and *localpop* is cleared. When all solutions in *arch* are selected to perform the search process, the final *pop* will be considered as the offspring archive.

G. PROCEDURE OF IMOTLBO

The details of the proposed IMOTLBO for solving the MOJRCSP are presented in Algorithm 2. The algorithm starts with a class of PS students, where PS is the number of students in the class. Each student corresponds to a solution of the studied problem, and has its own knowledge level.

Some of them will have better performance than others by execute class preview phase. Because there are two objectives to be considered, the Pareto ranking is applied to select the teachers (non-dominated solutions) which are saved into a teacher team (archive). Then, all of the students begin to study continuously until the stopping criterion is satisfied.

At each iteration, students learn from one teacher and communicate with other students, this step are realized through teaching phase and learning phase where PTL crossover is performed. After the communication in class, each student tries to improve his performance by using self-studying phase, where four neighborhood structures are provided as mutate operator. When all the students have completed their learning tasks in this iteration, the teacher team will be updated under the Pareto dominance relationship.

Subsequently, all teachers start their further study, in order to further improve their performance. In the teacher promotion phase, the proposed local search is implemented to all solutions in archive, and generates an offspring archive with a set of non-dominated solutions. This step can speed up the convergence of IMOTLBO. When the algorithm is terminated, the teacher solutions in archive are considered as the output of algorithm.

Algorithm 2 The Pseudo-Code of IMOTLBO

```
Input: Data of problem
Output: The best individuals
Begin
    Initialize the population (Class preview phase)
    Evaluate the fitness (Calculation of objective values)
    Identify the non-dominated solutions (Teacher selection)
    While stopping criterion is not satisfied DO
        Crossover between students and teachers (Teaching phase)
        Crossover between students (Learning phase)
        Mutation for students (Self-studying phase)
        Update teachers
        Local search for teachers (Teacher promotion phase)
    End While
    Output the teachers
End
```

IV. COMPUTATIONAL RESULTS AND COMPARISONS

A. EXPERIMENTAL SETUP

To evaluate the performance of the proposed IMOTLBO algorithm, experimental evaluation and comparison with other methods were performed. Two sets of problem instances are considered:

- The first data set (instance 1-12) is provided by Caumond *et al.* [7].
- The second data set (instance 13-30) is generated randomly by the protocol.

The Caumond benchmark set includes 12 small sized instances with size ranging from 3 jobs and 12 operations to 5 jobs and 18 operations, the number of machines is equal to 5. The generated data set is composed of 18 medium-large sized instances with size ranging from 3 jobs, 12 operations and 5 machines to 20 jobs, 110 operations and 10 machines. In this study, the processing time p_{ji} is randomly generated between a uniform distribution [1, 50]. Loaded trip duration τ_{rl} and empty trip duration σ_{rl} are generated in the same way as [7]. Loading time ε_c and unloading time ε_d are set to 1. Inspired by Gao *et al.* [24], the following formulas are used to generate the earliest due date and latest due date, respectively.

$$a_j = (1 + \frac{T \times n}{m}) \times \sum_{i=1}^{n_j} p_{ji} \tag{20}$$

$$b_j = (2 + \frac{T \times n}{m}) \times \sum_{i=1}^{n_j} p_{ji} \tag{21}$$

where T is a parameter that is equal to 0.3, n is the number of jobs, m is the number of machines.

B. MEASURE METRICS

The comparison of results of multiobjective optimization is different to the single objective method, because there is a set of non-dominated solutions, instead of a single solution. Several conditions should be respected for the search results of the approximated method: the distance to the absolute pareto optimal front is to be minimized, the diversity of the obtained solutions should be maximized and the spread of the obtained solutions should also be maximized. The following widely used performance measures are applied in this paper:

- Number of non-dominated solutions

This performance measure counts the total number of non-dominated solutions of the compared algorithms.

- μ_d-distance of Dugardin *et al.* [34]

This measure computes the size of distance between two non-dominated fronts. Let A and B be two non-dominated fronts to be compared and which are provided by different methods. n_A and n_B are the number of non-dominated solutions in front A and front B respectively. Then the μ_d-distance is defined as:

$$\mu_d = \frac{\frac{1}{n_B}\sum_{i=0}^{n_B} d_i}{\sqrt{(f_{1max} - f_{1min})^2 + (f_{2max} - f_{2min})^2}} \tag{22}$$

where, $D = \sqrt{(f_{1max} - f_{1min})^2 + (f_{2max} - f_{2min})^2}$ is the cross of the corresponding rectangle, d_i is the distance between a solution of A and its orthogonal projection on B. The value of d_i is negative when the solution i is below the front B (or the extrapolated front), positive otherwise. This measure is the proportional improvement to the diagonal of the rectangle. It shows if front A is closer to the optimal solutions than front B. The more negative value of μ_d is, the more A is better than B.

- Zitzler measure [35]

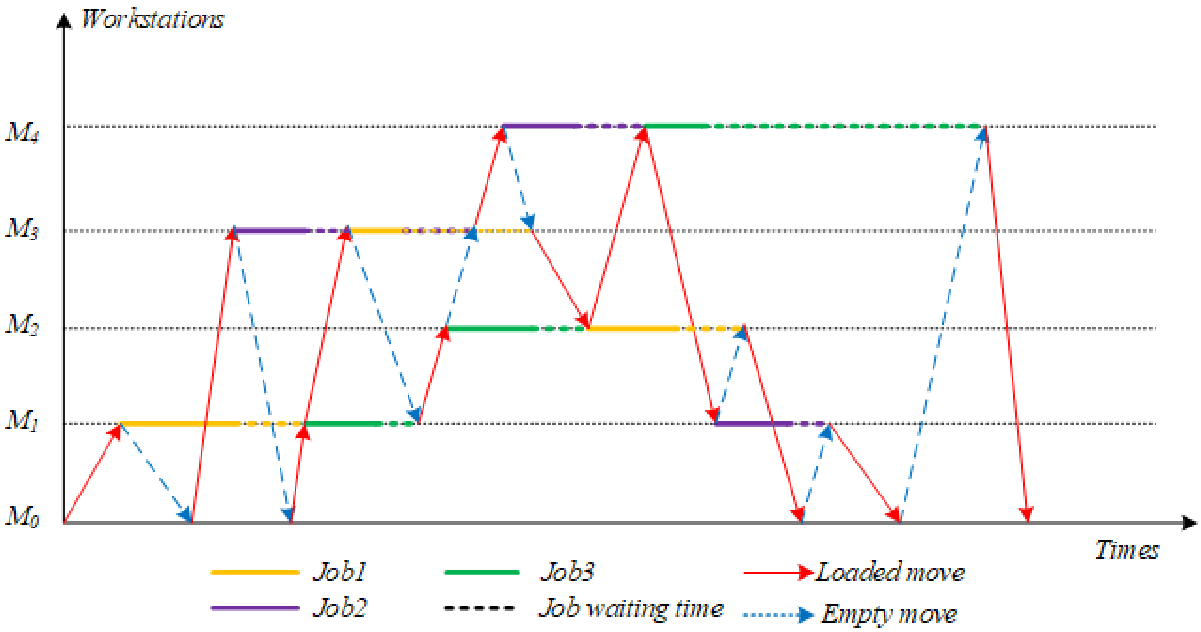

FIGURE 8. Example of gantt chart for small sized problem.

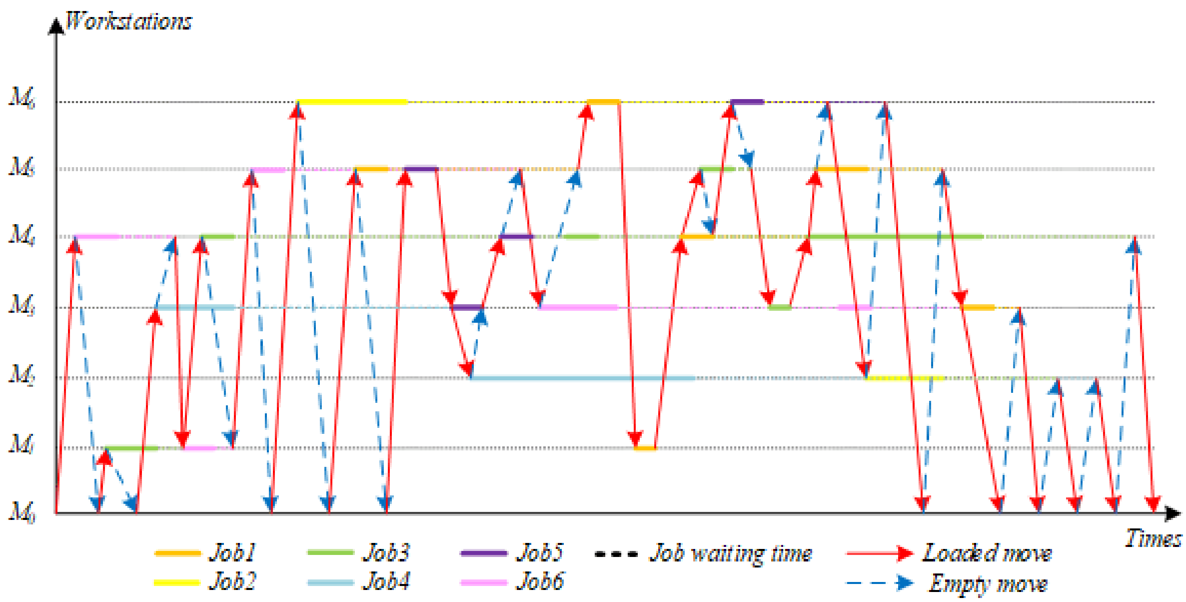

FIGURE 9. Example of gantt chart for medium sized problem.

According to the Zitzler measure, C_1 computes the ratio of solutions in A which are dominated by at least one solution in B and vice versa. C_1 is defined by

$$C_1 = \frac{|\{a \in A | \exists b \in B : b \succ a\}|}{|A|} \tag{23}$$

where a and b are the non-dominated solutions in the set A and B respectively.

C. RESULTS AND COMPARISONS

In order to demonstrate the performance of the proposed IMOTLBO algorithm, it is compared with several well-known and recently published algorithm, such as NSGA-II presented by Deb *et al.* [23], the multi-objective particle swarm optimization (MOPSO) presented by Sha and Hsu [25] and the Pareto-based grouping discrete harmony search algorithm (PGDHS) of Gao *et al.* [24]. All algo-

TABLE 1. IMOTLBO and NSGA-II of performance comparison.

Index	J	M	S	μ_d^*	$\mu_d^\#$	μ_d^b	C_1^*	$C_1^\#$	C_1^b	C_2^*	$C_2^\#$	C_2^b	n_1^*	n_2^*	CT_1	CT_2
1	3	5	12	-0.005	0.0001	-0.019	0.014	0.0006	0	0.337	0.014	0.526	22.3	16.2	3.2	4.2
2	4	5	15	-0.024	0.0001	-0.047	0.010	0.0006	0	0.816	0.0299	1	20.4	10.8	3.5	4.3
3	3	5	12	-0.038	0.0005	-0.086	0.029	0.0019	0	0.783	0.0210	1	29.1	9.6	3.4	4.5
4	3	5	12	-0.003	0.0001	-0.008	0.021	0.0007	0	0.251	0.0190	0.5	35.7	23.1	3.6	4.1
5	5	5	18	-0.090	0.0227	-0.490	0.113	0.0271	0	0.708	0.1045	1	5.7	2.8	3.4	4.6
6	4	5	15	-0.008	0.0001	-0.018	0.077	0.0050	0	0.482	0.0340	0.867	20	14.3	3.8	4.3
7	4	5	15	-0.028	0.0005	-0.085	0.055	0.0052	0	0.696	0.0275	1	16.2	9.8	3.5	4.4
8	4	5	15	-0.146	0.0140	-0.367	0.044	0.0107	0	0.796	0.0497	1	6.7	4.2	3.3	4.7
9	5	5	18	-0.010	0.0001	-0.019	0.079	0.0019	0	0.570	0.0168	0.846	31.7	15.0	3.9	4.5
10	5	5	18	-0.022	0.0003	-0.073	0.041	0.0045	0	0.767	0.0525	1	21.8	10.8	3.8	4.4
11	4	5	15	-0.073	0.0247	-0.458	0.089	0.0273	0	0.680	0.1250	1	5.2	4	3.7	4.6
12	5	5	18	0.004	0.0057	-0.031	0.022	0.0009	0	0.623	0.0288	0.941	34.4	18.9	3.7	4.8
13	3	5	12	-0.060	0.0057	-0.249	0.058	0.0102	0	0.813	0.0435	1	9.9	6.6	4.2	5.5
14	3	5	12	-0.033	0.0005	-0.074	0.015	0.0009	0	0.766	0.0349	1	37.2	9.9	4.4	5.5
15	4	5	15	-0.065	0.0011	-0.141	0.002	0.0001	0	0.901	0.0098	1	29.3	11	4.3	5.4
16	4	5	15	-0.056	0.0034	-0.180	0.061	0.0191	0	0.855	0.0585	1	28.8	9.1	4.3	5.3
17	5	5	18	-0.059	0.0008	-0.146	0.020	0.0014	0	0.873	0.0224	1	29.6	10.5	4.2	5.7
18	5	5	18	-0.048	0.0011	-0.139	0.051	0.0103	0	0.858	0.0242	1	26.3	10.2	4.3	5.6
19	6	5	23	-0.054	0.0011	-0.120	0.010	0.0004	0	0.941	0.0076	1	29.2	10.3	4.8	7.1
20	8	5	30	-0.076	0.0054	-0.232	0.008	0.0005	0	0.913	0.0256	1	32.1	11.9	4.9	7.4
21	8	5	30	-0.021	0.0001	-0.049	0.018	0.0012	0	0.650	0.0276	0.909	20.5	11.3	4.8	7.3
22	6	5	23	-0.011	0.0001	-0.028	0.007	0.0002	0	0.436	0.0366	0.765	31.0	16.4	5.2	7.4
23	6	7	31	-0.214	0.0141	-0.459	0.003	0.0002	0	0.988	0.0030	1	21.3	9.1	5.6	7.5
24	6	7	31	-0.017	0.0007	-0.096	0.077	0.0081	0	0.711	0.0210	1	12.8	6.2	6.6	7.8
25	10	7	47	-0.017	0.0002	-0.048	0.098	0.0086	0	0.648	0.0364	1	20.0	13.9	8.5	10.2
26	10	7	47	-0.026	0.0002	-0.059	0.046	0.0031	0	0.780	0.0356	1	25.5	13.3	8.7	10.5
27	10	7	56	-0.025	0.0002	-0.056	0.050	0.0479	0	0.870	0.0159	1	27.7	10.8	9.4	11.5
28	15	10	72	-0.031	0.0002	-0.077	0.023	0.0016	0	0.829	0.0114	1	22.2	13.7	9.9	12.0
29	15	10	83	-0.023	0.0001	-0.041	0.036	0.0030	0	0.760	0.0207	1	23.9	12.7	9.8	12.3
30	20	10	110	-0.042	0.0003	-0.090	0.015	0.0005	0	0.818	0.0131	1	34.0	11.8	10.0	12.7

rithms were coded in C++, and tested in a computer with Intel I7-8700 3.2 GHz CPU and 16GB RAM. Each instance runs for 20 replication. To ensure the fairness of comparison, the population size and number of iterations of four algorithms are set to 40 and 100. The comparison results are shown in Table 1-3.

In these tables, J is the number of jobs, M represents the number of machines and S is the numbers of total operations of the tested instance. The values of μ_d^*, $\mu_d^\#$ and μ_d^b represents the mean value, the variance and the best value of μ_d-distance. The next six values represent the mean value, the variance and the best value of C_1 and C_2 respectively.

TABLE 2. IMOTLBO and PGDHS of performance comparison.

Index	J	M	S	μ_d^*	$\mu_d^\#$	μ_d^b	C_1^*	$C_1^\#$	C_1^b	C_2^*	$C_2^\#$	C_2^b	n_1^*	n_2^*	CT_1	CT_2
1	3	5	12	-0.087	0.0056	-0.243	0.007	0.0003	0	0.656	0.0408	1	22.3	8.7	3.2	6.9
2	4	5	15	-0.090	0.0038	-0.234	0	0	0	0.841	0.0291	1	20.4	6.8	3.5	6.8
3	3	5	12	-0.115	0.0089	-0.450	0.012	0.0007	0	0.883	0.0242	1	29.1	7.0	3.4	7.0
4	3	5	12	-0.023	0.0014	-0.143	0.007	0.0001	0	0.420	0.0441	0.875	35.7	12.1	3.6	7.3
5	5	5	18	-0.669	0.0090	-0.857	0	0	0	1	0	1	5.7	2.9	3.4	7.1
6	4	5	15	-0.068	0.0046	-0.262	0.028	0.0023	0	0.723	0.0409	1	20.0	8.8	3.8	6.9
7	4	5	15	-0.111	0.0067	-0.285	0.028	0.0025	0	0.859	0.0363	1	16.2	6.8	3.5	7.4
8	4	5	15	-0.524	0.0577	-0.874	0	0	0	1	0	1	6.7	3.2	3.3	7.5
9	5	5	18	-0.043	0.0031	-0.236	0.031	0.0039	0	0.712	0.0520	1	31.7	9.9	3.9	7.3
10	5	5	18	-0.145	0.0154	-0.546	0.020	0.0037	0	0.915	0.0216	1	21.8	6.4	3.8	7.4
11	4	5	15	-0.621	0.0387	-0.861	0	0	0	1	0	1	5.2	3.4	3.7	7.2
12	5	5	18	-0.058	0.0026	-0.245	0.020	0.0017	0	0.711	0.0231	0.909	34.4	9.5	3.7	7.6
13	3	5	12	-0.342	0.0519	-0.664	0.033	0.0160	0	0.901	0.0593	1	9.9	4.0	4.2	10.1
14	3	5	12	-0.148	0.0157	-0.529	0.013	0.0010	0	0.883	0.0283	1	37.2	9.8	4.4	10.3
15	4	5	15	-0.108	0.0135	-0.569	0	0	0	0.971	0.0063	1	29.3	8.5	4.3	10.3
16	4	5	15	-0.169	0.0120	-0.480	0	0	0	0.969	0.0051	1	28.8	7.2	4.3	10.4
17	5	5	18	-0.142	0.0060	-0.330	0.006	0.0002	0	0.971	0.0035	1	29.6	7.1	4.2	10.2
18	5	5	18	-0.078	0.0036	-0.232	0.009	0.0008	0	0.959	0.0100	1	26.3	7.1	4.3	10.5
19	6	5	23	-0.065	0.0030	-0.168	0.006	0.0002	0	0.964	0.0059	1	29.2	7.5	4.8	10.8
20	8	5	30	-0.066	0.0045	-0.239	0.033	0.0098	0	0.932	0.0150	1	32.1	8.8	4.9	12.5
21	8	5	30	-0.132	0.0067	-0.324	0.002	0.0001	0	0.939	0.0064	1	20.5	7.1	4.8	12.7
22	6	5	23	-0.089	0.0038	-0.247	0.008	0.0004	0	0.758	0.0392	1	31.0	9.0	5.2	13.2
23	6	7	31	-0.206	0.0125	-0.472	0.003	0.0002	0	0.954	0.0099	1	21.3	5.4	5.6	13.8
24	6	7	31	-0.232	0.0164	-0.487	0.007	0.0010	0	0.973	0.0065	1	12.8	4.8	6.6	14.5
25	10	7	47	-0.068	0.0037	-0.233	0.030	0.0031	0	0.794	0.0210	1	20.0	9.0	8.5	18.4
26	10	7	47	-0.056	0.0048	-0.317	0.037	0.0020	0	0.775	0.0300	1	25.5	8.9	8.7	19.6
27	10	7	56	-0.067	0.0055	-0.324	0.007	0.0005	0	0.940	0.0065	1	27.7	8.6	9.4	20.5
28	15	10	72	-0.069	0.0017	-0.139	0.019	0.0010	0	0.882	0.0226	1	22.2	7.6	9.9	21.3
29	15	10	83	-0.061	0.0033	-0.248	0.031	0.0041	0	0.816	0.0420	1	23.9	8.5	9.8	22.1
30	20	10	110	-0.084	0.0034	-0.247	0.009	0.0003	0	0.959	0.0043	1	34.0	6.9	10.0	22.5

n_1^* and n_2^* are mean values of the numbers of non-dominated solutions of two compared algorithm. CT_1 and CT_2 are mean values of computing times in seconds.

The first line of Table 1 is explained. The problem is defined with 3 jobs, 12 operations and 5 machines. The thirteen following columns show the comparison results between IMOTLBO and NSGA-II. Since each instance is tested for 20 times, the mean value of the μ_d-distance $\mu_d^* = -0.005$ and the variance $\mu_d^\# = 0.0001$ indicate that IMOTLBO is better than ($\mu_d^* < 0$) the NSGA-II. In a best situation, we have $\mu_d^b = -0.019$. The result shows $C_1^* = 0.014$, *i.e.* there are only 1.4% solutions provided by IMOTLBO

TABLE 3. IMOTLBO and MOPSO of performance comparison.

Index	J	M	S	μ_d^*	$\mu_d^\#$	μ_d^b	C_1^*	$C_1^\#$	C_1^b	C_2^*	$C_2^\#$	C_2^b	n_1^*	n_2^*	CT_1	CT_2
1	3	5	12	-0.103	0.0091	-0.264	0.003	0.0001	0	0.879	0.0578	1	22.3	5.4	3.2	5.1
2	4	5	15	-0.135	0.0139	-0.340	0	0	0	0.972	0.0046	1	20.4	5.2	3.5	4.8
3	3	5	12	-0.196	0.0124	-0.381	0	0	0	1	0	1	29.1	6.7	3.4	5.0
4	3	5	12	-0.056	0.0028	-0.157	0	0	0	0.784	0.1062	1	35.7	5.6	3.6	4.6
5	5	5	18	-0.599	0.0321	-0.882	0	0	0	1	0	1	5.7	2.4	3.4	5.3
6	4	5	15	-0.082	0.0100	-0.357	0	0	0	0.767	0.0776	1	20.0	6.7	3.8	5.2
7	4	5	15	-0.093	0.0113	-0.365	0.003	0.0002	0	0.876	0.0267	1	16.2	6.1	3.5	5.2
8	4	5	15	-0.459	0.0444	-0.735	0	0	0	1	0	1	6.7	3.1	3.3	5.1
9	5	5	18	-0.113	0.0057	-0.273	0.005	0.0002	0	0.919	0.0156	1	31.7	7.0	3.9	4.9
10	5	5	18	-0.163	0.0098	-0.353	0.007	0.0006	0	0.964	0.0084	1	21.8	5.6	3.8	5.0
11	4	5	15	-0.375	0.0871	-0.826	0.013	0.0030	0	0.950	0.0225	1	5.2	3.0	3.7	5.2
12	5	5	18	-0.077	0.0050	-0.211	0.006	0.0002	0	0.795	0.0688	1	34.4	6.8	3.7	5.1
13	3	5	12	-0.284	0.0439	-0.644	0	0	0	0.9833	0.0053	1	9.9	3.1	4.2	7.4
14	3	5	12	-0.185	0.0244	-0.577	0.001	0.0001	0	0.992	0.0006	1	37.2	9.4	4.4	7.3
15	4	5	15	-0.167	0.0138	-0.499	0	0	0	0.996	0.0004	1	29.3	9.2	4.3	7.2
16	4	5	15	-0.216	0.0233	-0.561	0	0	0	1	0	1	28.8	8.4	4.3	6.9
17	5	5	18	-0.200	0.0064	-0.390	0	0	0	0.992	0.0011	1	29.6	10.1	4.2	7.0
18	5	5	18	-0.160	0.0138	-0.499	0	0	0	0.982	0.0031	1	26.3	8.1	4.3	7.4
19	6	5	23	-0.136	0.0141	-0.437	0	0	0	1	0	1	29.2	8.3	4.8	8.2
20	8	5	30	-0.072	0.0055	-0.275	0	0	0	0.979	0.0031	1	32.1	9.1	4.9	8.1
21	8	5	30	-0.170	0.0093	-0.364	0	0	0	1	0	1	20.5	4.2	4.8	8.4
22	6	5	23	-0.150	0.0154	-0.461	0	0	0	0.880	0.0591	1	31.0	4.6	5.2	9.0
23	6	7	31	-0.214	0.0141	-0.459	0.003	0.0002	0	0.988	0.0030	1	21.3	4.4	5.6	9.1
24	6	7	31	-0.279	0.0272	-0.516	0.008	0.0006	0	0.955	0.0145	1	12.8	4.6	6.6	9.3
25	10	7	47	-0.088	0.0064	-0.345	0.009	0.0015	0	0.950	0.0103	1	20.0	5.8	8.5	11.1
26	10	7	47	-0.078	0.0031	-0.239	0	0	0	0.945	0.0208	1	25.5	5.9	8.7	12.8
27	10	7	56	-0.130	0.0120	-0.369	0	0	0	0.995	0.0005	1	27.7	7.5	9.4	13.9
28	15	10	72	-0.086	0.0051	-0.291	0	0	0	0.996	0.0003	1	22.2	6.7	9.9	14.2
29	15	10	83	-0.134	0.0110	-0.389	0.005	0.0003	0	0.955	0.0122	1	23.9	6.9	9.8	14.5
30	20	10	110	-0.139	0.0080	-0.318	0	0	0	1	0	1	34.0	6.6	10.0	15.8

that are dominated by at least one solution from NSGA-II. As C_1, $C_2^* = 0.337$ suggests that 37.7% of the solutions obtained by NSGA-II are dominated by at least one solution of the IMOTLBO fronts. The small variance values of C_1 and C_2 also prove the conclusion ($C_1^\# = 0.0006$ and $C_2^\# = 0.014$). The numbers of non-dominated solutions show that the IMOTLBO has obtained more non-dominated solutions than NSGA-II with respective mean values of 22.3 and 16.2. Having CT_1 less than CT_2 means that IMOTLBO computational time is less than NSGA-II one (3.2 seconds and 4.2 seconds). The meaning of each column in each comparison is the same.

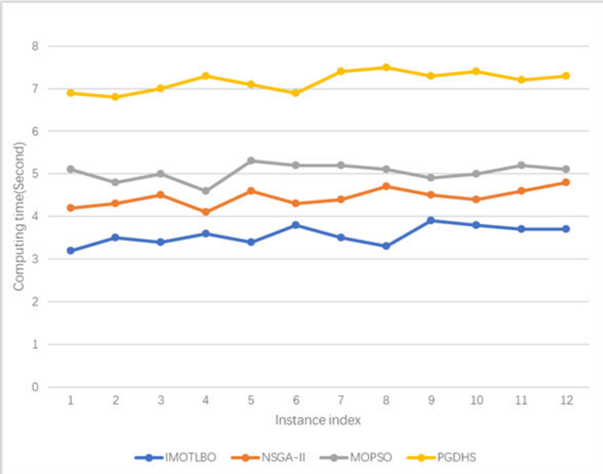

FIGURE 10. Computing times for instances (1)-(12).

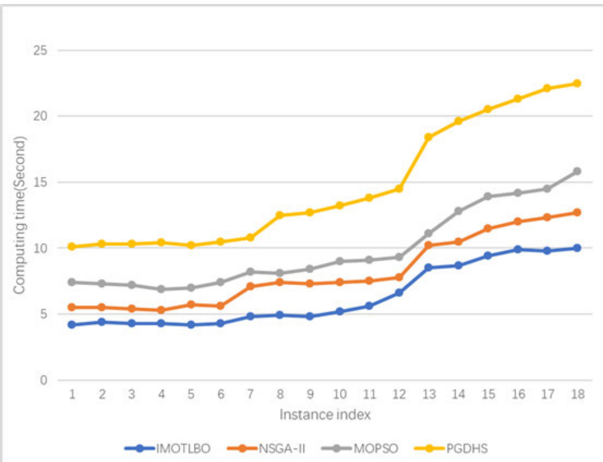

FIGURE 11. Computing times for instances (13)-(30).

In Table 1, we can get same conclusion through the data of other lines, we find that our proposed IMOTLBO algorithm outperforms NSGA-II. Moreover, a special case appears in line 12 when $\mu_d^* = 0004$, it is because the two fronts are too closed to each other (when one front overlaps another front, the d_i is alternatively positive and negative, the measure μ_d^* is not signficant [34]). In This case the μ_d^* dose not depict the relative position of the two fornts. The measures C_1/C_2 can be used to undertake the Comparison mission.

In Table2 and Table 3. The μ_d shows that IMOTLBO dominates PGDHS and MOPSO. The couple C_1/C_2 shows that fewer solution of the IMOTLBO are dominated by at least one solution of PGDHS and MOPSO ($C_1 < C_2$). IMOTLBO can produce more non-dominated solutions and require less computing time than ohter algorithms. By comparison, IMOTLBO is better than PGDHS and MOPSO. The comparison results fully prove the effectiveness, robustness and diversity of IMOTLBO algorithm.

Gantt chart helps us to observe the scheduling results intuitively, including the scheduling of operations on stations and the movements of handling robot. Two examples are shown in Figure 8 and Figure 9, corresponding to a small sized problem and a medium sized problem respectively. In these figures, X-axis represents times and Y-axis represents stations or machines.

Finally, we compared the computing times of all algorithm by graphics. The computing times of the four algorithms on Caumond benchmark are shown in Figure 10, Figure 11 shows the computing times on our generated instances. For all tested instances, the IMOTLBO algorithm shows its advantage in computing time.

V. CONCLUSION

In this paper, we focused on solving the multi-objective job-shop robotic cell scheduling problem (MOJRCSP), a Mixed Integer Programming model is proposed to model the studied problem. The objective is to minimize the makespan and total earliness and tardiness simultaneously. An improved multi-objective teaching learning based optimization algorithm is proposed, where 5 phased are implemented in order to improve the performance of algorithm. Through comparative studies with other multi-objective optimization algorithms, the proposed method outperforms the others.

REFERENCES

[1] S. Gürel, H. Gultekin, and V. E. Akhlaghi, "Energy conscious scheduling of a material handling robot in a manufacturing cell," *Robot. Comput.-Integr. Manuf.*, vol. 58, pp. 97–108, Aug. 2019.

[2] R. Abaza, D. D. Eun, M. Gallucci, I. S. Gill, M. Menon, A. Mottrie, and A. Shabsigh, "Robotic surgery for renal cell carcinoma with vena caval tumor thrombus," *Eur. Urol. Focus*, vol. 2, no. 6, pp. 601–607, Dec. 2016.

[3] R. Gerbers, M. Mücke, F. Dietrich, and K. Dröder, "Simplifying robot tools by taking advantage of sensor integration in human collaboration robots," *Procedia CIRP*, vol. 44, pp. 287–292, Mar. 2016.

[4] M. Foumani, A. Razeghi, and K. Smith-Miles, "Stochastic optimization of two-machine flow shop robotic cells with controllable inspection times: From theory toward practice," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101822.

[5] P. Yan, S. Q. Liu, T. Sun, and K. Ma, "A dynamic scheduling approach for optimizing the material handling operations in a robotic cell," *Comput. Oper. Res.*, vol. 99, pp. 166–177, Nov. 2018.

[6] A. Elmi and S. Topaloglu, "Cyclic job shop robotic cell scheduling problem: Ant colony optimization," *Comput. Ind. Eng.*, vol. 111, pp. 417–432, Sep. 2017.

[7] A. Caumond, P. Lacomme, A. Moukrim, and N. Tchernev, "An MILP for scheduling problems in an FMS with one vehicle," *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 706–722, Dec. 2009.

[8] A. Che, H. Hu, M. Chabrol, and M. Gourgand, "A polynomial algorithm for multi-robot 2-cyclic scheduling in a no-wait robotic cell," *Comput. Oper. Res.*, vol. 38, no. 9, pp. 1275–1285, Sep. 2011.

[9] M. Nattaf, S. Dauzère-Pérès, C. Yugma, and C.-H. Wu, "Parallel machine scheduling with time constraints on machine qualifications," *Comput. Oper. Res.*, vol. 107, pp. 61–76, Jul. 2019.

[10] G. Zhang and K. Xing, "Differential evolution metaheuristics for distributed limited-buffer flowshop scheduling with makespan criterion," *Comput. Oper. Res.*, vol. 108, pp. 33–43, Aug. 2019.

[11] D. Lei, "A Pareto archive particle swarm optimization for multi-objective job shop scheduling," *Comput. Ind. Eng.*, vol. 54, no. 4, pp. 960–971, May 2008.

[12] G. Wang, L. Gao, X. Li, P. Li, and M. F. Tasgetiren, "Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm," *Swarm Evol. Comput.*, vol. 57, Sep. 2020, Art. no. 100716.

[13] M. Kurdi, "An improved island model memetic algorithm with a new cooperation phase for multi-objective job shop scheduling problem," *Comput. Ind. Eng.*, vol. 111, pp. 183–201, Sep. 2017.
[14] A. Udomsakdigool and V. Khachitvichyanukul, "Ant colony algorithm for multi-criteria job shop scheduling to minimize makespan, mean flow time and mean tardiness," *Int. J. Manage. Sci. Eng. Manage.*, vol. 6, no. 2, pp. 116–122, Jun. 2011.
[15] H. Mokhtari and A. Hasani, "An energy-efficient multi-objective optimization for flexible job-shop scheduling problem," *Comput. Chem. Eng.*, vol. 104, pp. 339–352, Sep. 2017.
[16] F. Arnold and K. Sörensen, "Knowledge-guided local search for the vehicle routing problem," *Comput. Oper. Res.*, vol. 105, pp. 32–46, May 2019.
[17] R. Tavakkoli-Moghaddam, M. Azarkish, and A. Sadeghnejad-Barkousaraie, "A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 10812–10821, Sep. 2011.
[18] G. Moslehi and M. Mahnam, "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search," *Int. J. Prod. Econ.*, vol. 129, no. 1, pp. 14–22, Jan. 2011.
[19] R. Zhang, P.-C. Chang, S. Song, and C. Wu, "Local search enhanced multi-objective PSO algorithm for scheduling textile production processes with environmental considerations," *Appl. Soft Comput.*, vol. 61, pp. 447–467, Dec. 2017.
[20] Q. Luo, Q. Deng, G. Gong, L. Zhang, W. Han, and K. Li, "An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers," *Expert Syst. Appl.*, vol. 160, Dec. 2020, Art. no. 113721.
[21] M. Kumar, M. L. Mittal, G. Soni, and D. Joshi, "A hybrid TLBO-TS algorithm for integrated selection and scheduling of projects," *Comput. Ind. Eng.*, vol. 119, pp. 121–130, May 2018.
[22] V. Toğan, "Design of planar steel frames using teaching–learning based optimization," *Eng. Struct.*, vol. 34, pp. 225–232, Jan. 2012.
[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Aug. 2002.
[24] K. Z. Gao, P. N. Suganthan, Q. K. Pan, T. J. Chua, T. X. Cai, and C. S. Chong, "Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling," *Inf. Sci.*, vol. 289, pp. 76–90, Dec. 2014.
[25] D. Y. Sha and C.-Y. Hsu, "A new particle swarm optimization for the open shop scheduling problem," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3243–3261, Oct. 2008.
[26] V. K. Patel and V. J. Savsani, "A multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO)," *Inf. Sci.*, vol. 357, pp. 182–200, Aug. 2016.
[27] Y. Xu, L. Wang, S.-Y. Wang, and M. Liu, "An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time," *Neurocomputing*, vol. 148, pp. 260–268, Jan. 2015.
[28] S. Chinta, R. Kommadath, and P. Kotecha, "A note on multi-objective improved teaching-learning based optimization algorithm (MO-ITLBO)," *Inf. Sci.*, vol. 373, pp. 337–350, Mar. 2011.
[29] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.
[30] M. Nawaz, E. E. Enscore, Jr., and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, Jan. 1983.
[31] O. Holthaus and C. Rajendran, "Efficient dispatching rules for scheduling in a job shop," *Int. J. Prod. Econ.*, vol. 48, no. 1, pp. 87–105, Jan. 1997.
[32] Q.-K. Pan, M. Fatih Tasgetiren, and Y.-C. Liang, "A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem," *Comput. Oper. Res.*, vol. 35, no. 9, pp. 2807–2839, Sep. 2008.
[33] P. Hansen, N. Mladenović, and J. A. M. Pérez, "Variable neighborhood search: Methods and applications," *4OR*, vol. 6, no. 4, pp. 319–360, Dec. 2008.
[34] F. Dugardin, F. Yalaoui, and L. Amodeo, "New multi-objective method to solve reentrant hybrid flow shop scheduling problem," *Eur. J. Oper. Res.*, vol. 203, no. 1, pp. 22–31, May 2010.
[35] E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithm: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

XIAOHUI LI (Member, IEEE) was born in Xi'an, Shaanxi, China, in 1982. He received the B.Eng. degree from the Xi'an University of Technology, China, in 2004, the M.Eng. degree from Paul Sabatier University, France, in 2008, and the Ph.D. degree from the University of Technology of Troyes, France, in 2011. Since 2012, he has been a Lecturer with the School of Electronics and Control Engineering, Chang'an University. His research interests include optimization algorithm, scheduling problem, vehicle routing problem, UAV path planning, and multi-objective optimization.

XI YANG was born in Baoji, Shaanxi, China, in 1994. She received the bachelor's degree in measurement and control technology and instrumentation from Xi'an Aeronautical University, in 2016. She is currently pursuing the master's degree with Chang'an University. Her main research interests include intelligent optimization algorithm and workshops scheduling optimization problem.

YI ZHAO (Member, IEEE) received the M.Eng. degree from Pierre and Marie Curie University (University Paris VI), in 2009, and the Ph.D. degree from the University of the South, Toulon-Var, in 2013. Since 2014, he has been with the School of Electronics and Control Engineering, Chang'an University. His current research interests include machine learning and sensor networks.

YING TENG was born in Shehong, Sichuan, China, in 1967. She received the M.S. degree in economics and management from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, in 1989, and the Ph.D. degree in management science and engineering from Southwest Jiaotong University, Chengdu, in 2006. From 2002 to 2008, she was an Associate Professor with the School of Economics and Management, UESTC, where she has been a Professor since 2008. She is currently the Director of the Center for Israeli Studies, UESTC. Her research interests include business model innovation under the background of new economy, platform governance model, access pricing, third-degree price discrimination with network externalities, and telecommunication industry.

YUAN DONG was born in Xi'an, Shaanxi, China, in 1987. She received the B.E. degree in electronic and information engineering from Northwestern Polytechnical University, in 2009, and the master's and Ph.D. degrees in computer science from the University of Technology of Troyes, France, in 2013. Since 2014, she has been doing her research and teaching with Chang'an University. She has authored over ten technical papers in refereed journals and conference proceedings. Her main research interests include machine learning, pattern recognition, wireless heterogeneous networks, especially in study of image processing and decision systems.