Scheduling of Dynamic Multi-Objective Flexible Enterprise Job-Shop Problem Based on Hybrid QPSO

Wei Chen1,2, Hong Yang3*, Yifei Hao4
1Research Center for Enterprise Management, Chongqing Technology and Business University, Chongqing 400067, China; 2School of Management, Chongqing Technology and Business University, Chongqing 400067, China; 3School of Geography and Tourism, Chongqing Normal University, Chongqing 401331, China; 4College of Mathematics and Statistics, Chongqing Technology and Business University, Chongqing 400067, China

Corresponding author: Hong Yang (e-mail: cqnuyh@163.com).

The authors acknowledge funding from the National Social Science Foundation of China (Project No. 18CGL003), as well as the contributions from all partners of the mentioned projects. Besides, Yang Hong is the corresponding author and can be contacted at: cqnuyh@163.com.

ABSTRACT In view of the importance of flexible job-shop scheduling problem (FJSP) in actual production, this paper constructs a mathematical model of fuzzy FJSP and then proposes a mixed quantum algorithm based on local optimization strategy and improved optimization rotation angle. For improving the production process, a double chain coding method was designed with two gene chains, which respectively represent the machine selection and the process sequencing. Next, the hybrid quantum particle swarm optimization (QPSO) was introduced to ensure the scheduling efficiency. Finally, the prototype system of the proposed strategy was simulated by using some actual examples. The results show that the proposed algorithm can quickly form an adjusted plan that has minimal difference from the original plan.

INDEX TERMS Flexible job-shop scheduling, quantum particle swarm optimization, quantum gate rotation angle, double chain quantum coding

I. INTRODUCTION
As the key to production management, scheduling directly bears on the internal resource allocation and scientific management of enterprises [1-4]. During the production, much time is consumed in processes irrelevant to actual operations. The waste of time can be minimized through proper scheduling, leading to lower cost and better efficiency. As a result, effective scheduling ensures the orderliness, controllability and efficiency of job-shop production.

The purpose of job-shop scheduling is to optimize performance indices through proper arrangement of work processes. The optimization process must consider the following constraints: For n jobs to be processed on m machines, each job can be treated with multiple processes and should be processed by the machines following a certain order, while each machine can process the same job with a unique set of processes [5].

Job-shop scheduling problems can be divided into different categories. By processing features, these problems fall into static scheduling and dynamic scheduling; By the types of jobs and job-shops, these problems fall into open shop scheduling, flow shop scheduling, single machine scheduling, parallel machine scheduling and classic job-shop scheduling [6-7].

To mimic the actual production, the classic job-shop scheduling has been extended to the flexible job-shop scheduling (FJSP) [8], which eliminates the uniqueness of processes on each machine by assuming that a process can be implemented on one of the machines. The work flow of the FJSP is illustrated as Figure 1 below.

In the existing studies, the FJSP covers both single-objective problems and multi-objective problems [9-11]. In multi-objective problems, fuzzy FJSP and dynamic FJSP are relatively complex and close to actual production [12]. At present, the FJSP research is focused on the comprehensive application of various intelligent algorithms, and the evaluation of scheduling performance based on time.
In general, there are four types of FJSP optimization algorithms, namely, direct assignment algorithms, accurate algorithms, classical heuristic algorithms and sub-optimal heuristic algorithms. Specifically, the direct assignment algorithms, pursuing the earliest delivery, shortest processing time or longest processing time, can approximate the optimal solution in a short time; the accurate algorithms, namely, cutting algorithms, branch and bound method, dynamic programming method and network-flow method, abstracts constraints and objective functions using the mathematical programming theory; the classical and sub-optimal heuristic algorithms (e.g. bacterial algorithm, bee colony algorithm, genetic algorithm, ant colony algorithm and artificial neural network) can obtain the optimal solution according to the features of the scheduling problem.

II. LITERATURE REVIEW

FJSP, a typical NP-hard problem, contains two sub-problems: the selection of machines for job processing and the sorting of the processes. The existing FJSP research mainly relies on the splitting method and the integration method [13]. According to resource constraints, the FJSP is either a total FJSP or a partial FJSP. The former allows all jobs to be processed on any machine, while the latter requires that at least one process must be implemented on only one of the machines.

As mentioned before, the FJSP covers both single-objective problems and multi-objective problems. The single-objective FJSP only has one optimization target. Kacem solved the single-objective FJSP through minimizing the makespan with genetic algorithm [14]. Zhang et al. combined a hybrid genetic algorithm and multiple crossover and mutation to minimize the penalty for solving the FJSP [15]. Based on improved ant colony algorithm, Wang et al. minimized the idle time of machines through pheromone adjustment, and proposed a machine selection strategy that quickly converges to the global optimal solution [16].

By contrast, fuzzy FJSP and dynamic FJSP are relatively complex in multi-objective problems. The processes in multi-objective FJSP may lead to multiple alternative paths and expand the feasible solution space. Chen et al. created an algorithm based on binary particle swarm optimization algorithm and successfully approximated the optimal solution with the algorithm [17]. Jin et al. integrated the convergence of particle swarm optimization with quantum theory, creating a particle swarm optimization algorithm with quantum 17 to solve the FJSP [18]. Zhang et al. proposed a multi-objective flexible batch scheduling algorithm based on multi-population genetic algorithm [19].

The development of fuzzy mathematics has opened a new direction for the solution of fuzzy FJSP. Yin et al. invented a fuzzy number method based on the delivery time, aiming to solve two-machine scheduling and parallel machine scheduling in open-loop job-shop [20]. Wang et al. tackled parallel machine scheduling by solving fuzzy processing time with tabu search algorithm [21].

With the growing demand for product diversification, more and more attention has been paid to FJSP with fuzzy processing time and delivery time. Sakawa et al. developed an efficient genetic algorithm for single-objective FJSP with fuzzy delivery time and fuzzy processing time [22]. Geng et al. systematically explored fuzzy single-objective job-shop scheduling and fuzzy flow-shop scheduling, and recommended the fuzzy scheduling strategies based on dispatching rules, sequence and heuristic algorithm [23].

In actual production, the scheduling problems are constantly changing rather than stick to the plan. Unexpected situations may occur from time to time, such as temporary cancellation of orders, emergency insertion or machine failure, which calls for dynamic job-shop scheduling. Despite the high complexity, dynamic scheduling has become a research hotspot because it is close to the actual production. Chu et al. defined the boundary between static scheduling and dynamic scheduling by clustering method [24]. Geyik et al. studied the adaptive scheduling strategies of dynamic scheduling [25], and categorized these strategies into cycle-driven rolling strategy, event-driven rolling strategy and hybrid strategy driven by both cycle and event. Liu et al. proposed a new optimization genetic algorithm to solve FJSP, using sequence coding and active scheduling coding [26]. Yu et al. discussed the multi-collaborative job scheduling based on parallel co-evolutionary genetic algorithm [27]. Inspired by multi-population genetic algorithm, Liu et al. developed the multi-objective flexible batch scheduling algorithm, which uses fuzzy membership degree to describe the processing time and delivery time [28].

III. RESEARCH METHODS

A. MATHEMATICAL MODEL OF FJSP

The goals of multi-objective FJSP include minimizing the maximum makespan, the total cost, the penalty, the consumer dissatisfaction and the most loaded machine. Here, four objective functions are defined below for the problem.

The maximum makespan refers to the longest makespan in all processes of a job. It is an important indicator of the production efficiency and a key evaluator of the scheduling plan.

Minimizing the maximum makespan \( f_1: \)
\[ f1 = min(F) = min \left[ \max \left( \sum_{m=1}^{M} F_m \right) \right] \]  
(1)

where \( F \) is the makespan of all machines.

\[ F_m = \sum_{i=1}^{N} \sum_{j=1}^{n_i} \left( S_{ijm} b_{ijm} + S_{ijm} t_{ijm} \right) \]

(2)

where, \( F_m \) is the total makespan of machine \( m \); \( b_{ijm}, t_{ijm} \) and \( S_{ijm} \) are respectively the start time, duration and end time of the implementation of process \( R_{ij} \) on machine \( m \).

The total cost includes raw material cost, machine loss cost, labour cost and inventory cost.

Minimizing the total cost \( f2 \):

\[ f2 = min(C) = \min \left[ \sum_{i=1}^{N} \left( M_i n_i + \sum_{j=1}^{n_i} \sum_{m=1}^{M} C_{ijm} S_{ijm} \right) \right] \]

(3)

where, \( C \) is the total cost of job \( i \); \( M_i \) is the raw material cost of job \( i \); \( C_{ijm} \) is the cost of implementing process \( R_{ij} \) on machine \( m \).

\[ C_{ijm} = (\mu_{ijm} + \nu_{ijm}) \]

(4)

where, \( \mu_{ijm} \) and \( \nu_{ijm} \) are respectively the labour cost and machine cost of implementing process \( R_{ij} \) on machine \( m \).

To minimize the average makespan, it is necessary to measure the scheduling efficiency, machine utilization and production efficiency.

Minimizing the average makespan \( f3 \):

\[ f3 = min \left( \frac{1}{N} \sum_{j=1}^{N} F_j \right) \]

(5)

In the course of scheduling, the most loaded machine bottlenecks the job-shop production.

Minimizing the most loaded machine \( f4 \):

\[ f4 = min \left( \max_{1 \leq m \leq M} \sum_{i=1}^{N} \sum_{j=1}^{n_i} t_{ijm} S_{ijm} \right) \]

(6)

Besides, the following constraints were defined for our problem:

Process constraint:

The processes of the same job must follow a fixed sequence, i.e. the \( j \)-th process of job \( i \) must be carried out after the completion of the previous (\( j-1 \)-th) process.

\[ \sum_{m=1}^{M} b_{ijm} S_{ijm} \geq \sum_{m=1}^{M} [(b_{i(j-1)m} - t_{i(j-1)m})] S_{i(j-1)m} \]

(7)

where, \( b_{ijm} \) is the start time of process \( R_{ij} \) on machine \( m \); \( S_{ijm} = S_{ij(j-1)m} = 1 \).

Machine constraint:

Only one process can be implemented on the same machine at a time. For process \( R_{ij} \), if \( \exists S_{ijm} = 1 \) at the time \( t \) (\( t>0 \)), then \( S_{sym} \neq 1 \) (when \( i=x, j \neq y \)).

Continuity constraint:

Process \( R_{ij} \) cannot be interrupted during processing.

\[ c_{ijm} = \begin{cases} \max \left[ (c_{i(l-1)m} + b_{ijm} + t_{ijm}), j > 1 \right] & \text{for } j > 1 \\ (b_{ijm} + t_{ijm}) & \text{for } j = 1 \end{cases} \]

(8)

where \( c_{ijm} \) is the makespan of process \( R_{ij} \).

B. QUANTUM PARTICLE SWARM OPTIMIZATION ALGORITHM BASED ON IMPROVED ROTATION ANGLE

With a few simple rules, the particle swarm optimization (PSO) algorithm converges to the optimal solution quickly by iteration. To improve the evolution of quantum individuals and the convergence to the optimal solution, an improved QPSO algorithm was designed based on dynamic rotation angle for multi-objective FJSP.

In the QPSO, a particle in the quantum space is determined by a probability density function and its motion by a wave function rather than velocity and position vectors. The probability density function can be expressed as the following Schrodinger equation:

\[ |\phi(x)|^2 = \frac{1}{L} \exp \left( -2 \frac{||p - x||}{L} \right) \]

(9)

where, \( L(t+1) = 2\alpha \times |p - x(t)| \) is the search scope of particles.

The quantum state can be transformed into a normal state by Monte-Carlo method [29-30]. The particle update equation can be expressed as:

\[
\left\{ \begin{array}{l}
    p_t = \frac{a \ast p_{best_t} + b \ast g_{best_t}}{a + b} \\
    m_{best_t} = \frac{\sum_{r=1}^{M} p_{best_t}(r)}{M}
\end{array} \right. \\
    x_{t+1} = p_t \pm \beta \ast |m_{best_t} - x_t| \ast \ln \frac{1}{\eta}
\]

where, \( p_{best_t}(r) \) is the optimal solution of the position vector determined by the \( r \)-th (\( 1 \leq r \leq M \)) particle; \( m_{best_t} \) is the global optimal solution for all particle position vectors; \( p_t \) is a random position vector between \( m_{best_t} \) and \( g_{best_t} \); \( M \) and \( t \) are the number of particles and the number of iterations, respectively; \( \beta \) is the creative coefficient function to control the convergence speed; \( a, b \) and \( u \) are random numbers between 0 and 1.

On the upside, the QPSO boasts only a few parameters and a strong global optimization ability; on the downside, it is prone to the local optimum trap due to the loss of diversity of the population on the later stage. In the QPSO, the particle swarm is updated by the quantum rotation gate. Thus, a dynamic method was created to adjust the rotation angle of the quantum gate and ensure that the quantum individuals are updated independently.

The common definition of quantum rotation gate is:
where, \( \delta \theta = D(\alpha, \beta) \times \theta \); \( D(\alpha, \beta) \) is the rotation direction that constrains the algorithm convergence. The dynamic method can be expressed as:

\[
\theta_i = D(\alpha_i, \beta_i) \Delta \theta_i
\]

where, \( D(\alpha_i, \beta_i) \) is the rotation direction of rotation angle; \( \Delta \theta_i \) is the angular step length of rotation angle. The value of \( \Delta \theta_i \) should be increased if the objective function changes little. The rotation angle can be determined by the look-up table below.

### TABLE I

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( b_i )</th>
<th>( f(x) \geq f(b) )</th>
<th>( \Delta \theta_i )</th>
<th>( D(\alpha_i, \beta_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>T</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>T</td>
<td>cg</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>F</td>
<td>cg</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>T</td>
<td>cg</td>
<td>+1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>F</td>
<td>cg</td>
<td>+1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>T</td>
<td>cg</td>
<td>+1</td>
</tr>
</tbody>
</table>

In Table I, \( x_i \) is the i-th bit of the binary solution, \( b_i \) is the i-th bit of the optimal solution, \( f(x) \) is the adaptive function and \( cg \) is the convergence velocity coefficient:

\[
\text{cg} = 0.04\pi(1 - c \frac{n}{s\text{gen} + 1})
\]

where, \( n \) is the current number of iterations; \( c \) is a constant between 0 and 1; \( s\text{gen} \) is the maximum number of iterations.

The steps of the improved QPSO are as follows:

#### Step 1: Initialize the number of jobs \( n \), the number of machines \( m \), and the reference processing time of each process. Each particle corresponds to an \( n+m-1 \) dimensional vector, that is, a solution to the problem.

#### Step 2: Evaluate the initial fitness of each particle, and record the optimal position and optimal fitness of each particle.

#### Step 3: Perform non-dominated scheduling and calculate the crowding distance of the obtained plans according to different objectives.

#### Step 4: Select two near optimal solutions and compare their non-inferior solutions. Select the individual with the higher grade. In this way, generate the new population \( S \).

#### Step 5: Determine the degree and direction of the quantum gate rotation angle. Adjust the quantum rotation angle dynamically according to the formula of convergence velocity coefficient.

#### Step 6: Judge whether the convergence condition is satisfied, or whether the maximum number of iterations is reached. If yes, jump to Step 7; otherwise, return to Step 3.

#### Step 7: Output the value of performance indices of the current optimal scheduling plan \( X_k \) and global near optimal scheduling plan.

### IV. Hybrid QPSO Algorithm for Dynamic FJSP

In dynamic scheduling, the multiple objectives should be realized without significantly changing the original scheduling plan. Based on the improved QPSO, this section proposes a double chain coding method to effectively solve the dynamic FJSP, in which the two gene chains respectively represent machine distribution and process sequence.

#### A. Mathematical model of dynamic FJSP

Suppose there are \( n \) jobs at the beginning, and \( n \) additional jobs after the initial scheduling. A total of \( m \) machines is ready for selection. Let \( k, i, i' \) and \( j \) be the machine, initial job, additional job and process, respectively, and \( R_{ij} \) represents the j-th process of job i.

Then, the objective functions were established as follows:

\[
t_{ijk}: \text{processing time of process } R_{ij} \text{ on machine } k;\]

\[
\begin{align*}
S_{ijk} &= 1, & \text{Process } R_{ij} \text{ is implemented on machine } k \\
S_{ijk} &= 0, & \text{Process } R_{ij} \text{ is not implemented on machine } k \\
S_{ij} &= 1, & \text{Process } R_{ij} \text{ is implemented on machine } k \\
S_{ij} &= 0, & \text{Process } R_{ij} \text{ is not implemented on machine } k
\end{align*}
\]

\[
\begin{align*}
b_i: & \text{The start time of the } j-\text{th process for the initial job } i. \\
b_{i'}: & \text{The start time of the } j-\text{th process for the initial job } i'. \\
b_j: & \text{The start time of the } j-\text{th process for the initial job } i. \\
b_{j'}: & \text{The start time of the } j-\text{th process for the initial job } i'.
\end{align*}
\]

The model mainly is used to solve the multi-objectives for scheduling efficiency and stability. The scheduling efficiency can be evaluated against the makespan and delay of the job through using the following formula:

\[
\text{Efficiency} = 5 \cdot (\max(F_n) - \min(b_n)) + 2 \cdot \sum \phi_n (F_n - DL_n)
\]

where, \( F_n \) is the makespan of job n; \( b_n \) is the start time of the process of job n; \( DL_n \) is the delivery time of job n. If \( F_n - DL_n > 0, \phi_n = 1; \) otherwise, \( \phi_n = 0. \)

The scheduling stability can be evaluated against the deviation penalty and the start time difference between the initial plan and the adjusted plan, using the following formula:

\[
\text{stability} = \sum \sum |\tau_{nm} - t_{nm}| + \sum \sum \Phi F(t_{nm} + \tau_{nm})
\]

where, \( \tau_{nm} \) is the start time of the process of job n on machine m in the adjusted plan; \( t_{nm} \) is the start time of the process of job n on machine m in the initial plan; \( \Phi F(\cdot) \) is the deviation penalty function; \( RT \) is the current processing time.
To enhance the stability, the scheduling stability formula can be rewritten as:

\[
\text{stability} = \frac{\sum \sum [t_{nm} - t_{nm}] + \sum \sum PF(t_{nm} + t_{nm} - 2RT)]}{\sum_{i=1}^{n} t_{nm} / n}
\]  

(16)

where, \( n_{1} \) is the number of processes for the job.

Thus, the objective function of dynamic FJSP can be defined as:

\[
\min(Z) = 5 \cdot (\max(F_{n}) - \min(b_{n})) + 2 \cdot \sum q_{n}(F_{n} - DL_{n}) + 4 \cdot \frac{\sum \sum [t_{nm} - t_{nm}] + \sum \sum PF(t_{nm} + t_{nm} - 2RT)]}{\sum_{i=1}^{n} t_{nm} / n}
\]

(17)

The constraints were defined as follows:

**Makespan:**

The makespan of any initial and adjusted process should not exceed the makespan of the job.

\[
C_{\max} \geq f_{ij} \quad i = 1,2, \ldots, n; j = 1,2, \ldots, j_{i}
\]

(18)

\[
C_{\max} \geq f'_{ij} \quad i = 1,2, \ldots, n'; j = 1,2, \ldots, j_{i'}
\]

(19)

where, \( f_{ij} \) and \( f'_{ij} \) are the makespans of processes \( R_{ij} \) and \( R'_{ij} \), respectively.

**Start time of process:**

The earliest start time of any process must be earlier than the adjusted start time:

\[
b_{ij} \geq mt_{k} \cdot S_{ijk} + RT \quad k = 1,2, \ldots, m; i = 1,2, \ldots, n; j = 1,2, \ldots, j_{i}
\]

(20)

\[
b'_{ij} \geq mt'_{k} \cdot S_{ijk} + RT \quad k = 1,2, \ldots, m; i' = 1,2, \ldots, n'; j = 1,2, \ldots, j_{i'}
\]

(21)

where, \( RT \) is the adjusted start time; \( mt_{k} \) is the idle time of machine \( k \) at the adjusted start time.

**Sequence:**

Process \( R_{ij} \) should follow the pre-set sequence:

\[
b_{ij} + t_{ij} \leq b_{i(j+1)} \quad i = 1,2, \ldots, n; j = 1,2, \ldots, j_{i} - 1
\]

(22)

\[
b'_{ij} + t'_{ij} \leq b'_{i(j+1)} \quad i' = 1,2, \ldots, n'; j = 1,2, \ldots, j_{i'} - 1
\]

(23)

**Machine:**

The start time of each process depends on the idle state of the selected machine. The process can be implemented only when the machine is idle.

\[
\sum_{k=1}^{m} \sum_{a=1}^{a_{k}} y_{kja} = 1 \quad i = 1,2, \ldots, n; j = 1,2, \ldots, j_{i}
\]

(24)

\[
\sum_{k=1}^{m} \sum_{a=1}^{a_{k}} y_{k'ja} = 1 \quad i = 1,2, \ldots, n'; j = 1,2, \ldots, j_{i'}
\]

(25)

\[
\sum_{k=1}^{m} S_{ijk} = 1 \quad i = 1,2, \ldots, n; j = 1,2, \ldots, j_{i}
\]

(26)

\[
\sum_{k=1}^{m} S'_{ijk} = 1 \quad i = 1,2, \ldots, n'; j = 1,2, \ldots, j_{i'}
\]

(27)

**B. HYBRID QPSO**

The QPSO based on double chain coding was proposed considering the two sub-problems of the initial and adjusted scheduling plans: machine selection and process sequencing. The QPSO was designed based on the quantum genetic algorithm (QGA), which is a combination of quantum evolutionary algorithm and genetic algorithm. The key to the QGA lies in the addition of quantum crossover and quantum variation after quantum rotation gate operation. The common approach to dynamic adjustment of quantum rotation gate method is as follows:

\[
[\alpha_{i} \beta_{i}^{T}] = [\cos(\theta_{i}) - \sin(\theta_{i}) \sin(\theta_{i}) \cos(\theta_{i})][\alpha_{i} \beta_{i}]
\]

(28)

where, \([\alpha_{i}, \beta_{i}]^{T}\) is the \(i\)-th qubit of chromosome; \(\theta_{i}\) is the rotation angle.

However, the QGA faces multiple problems in coding and decoding when applied to multi-objective complex optimization problems. As a result, the author put forward the improved double chain coding method by introducing a new compensation factor \(\gamma (\gamma \geq 1)\). Assuming that \(p_{i}\) is a quantum chromosome, the encoding plan for the \(i\)-th chromosome can be expressed as:

\[
p_{i} = [\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{im}] = [\cos(\gamma t_{i1}), \cos(\gamma t_{i2}), \ldots, \cos(\gamma t_{im})]
\]

(29)

\[
[p_{i} = [\sin(\gamma t_{i1}), \sin(\gamma t_{i2}), \ldots, \sin(\gamma t_{im})]
\]

(30)

where, \(\alpha_{i}\) and \(\beta_{i}\) must satisfy the normalized constraint \(|\alpha_{i}|^{2} + |\beta_{i}|^{2} = 1\).

Each chromosome contains two gene chains that respectively represent the machine allocation chain and process chain in the FJSP. Each gene chain corresponds to an optimal solution, and each chromosome may have two optimal solutions in the search space.

\[
p_{icos} = (\cos(t_{i1}), \cos(t_{i2}), \ldots, \cos(t_{im}))
\]

(30)

\[
p_{isin} = (\sin(t_{i1}), \sin(t_{i2}), \ldots, \sin(t_{im}))
\]

(31)

where, \(p_{icos}\) and \(p_{isin}\) are respectively the cosine solution and sine solution.

The hybrid QPSO based on double chain coding is implemented in the following steps:
Step 1: Initialize all parameters of jobs and machines, and generate particle position and velocity in quantum space by logistic equation.

Step 2: Apply double chain quantum coding to the initial solution, and express chromosome as machine gene chain and process gene chain.

Step 3: Calculate the objective function values of all particles in the population, with lbest_i being the position of the local optimum particle and gbest being the position of the particle corresponding to the smallest objective function value.

Step 4: Calculate mbest_i, and update the position of each particle.

Step 5: Calculate the objective function of each particle according to the adjusted target value, and update lbest_i and gbest_i.

Step 6: Check if the generated optimal plan outputs the required target values. If yes, jump to Step 5; otherwise, jump to Step 4.

Step 7: Output the optimal scheduling plan and the corresponding objective function value, and terminate the execution of the algorithm.

V. SIMULATION AND ANALYSIS

A. EXAMPLE ANALYSIS

A scheduling plan can be expressed as an n+1 dimensional particle, with n being the number of job, if the particles are coded in natural numbers. Taking n=6 for example, a particle (2,4,5,3,1,6, x) means the 2nd, 4th, 5th, 3rd, 1st and 6th jobs can be processed by the same machine. The proposed strategy was verified by the simulation test proposed by Kacem and Brandimarte [31-32].

The proposed hybrid QPSO algorithm was contrasted with AL+CGA, AIA, CPSO and HVNSA algorithms against two standard scales (8×8 and 10×10) of Kacem dataset. The maximum number of iterations was 100. The test results are listed in Table II below.

<table>
<thead>
<tr>
<th>n×m</th>
<th>AL+CGA</th>
<th>AIA</th>
<th>CPSO</th>
<th>HVNSA</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tx</td>
<td>Tx</td>
<td>CPU(s)</td>
<td>Tx</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>8×8</td>
<td>15</td>
<td>15</td>
<td>3.6</td>
<td>14</td>
<td>1.4</td>
</tr>
<tr>
<td>10×10</td>
<td>7</td>
<td>7</td>
<td>6.2</td>
<td>7</td>
<td>2.1</td>
</tr>
</tbody>
</table>

In Table II, n is the number of jobs, m is the number of machines, Tx is the makespan, and CPU(s) is the computing time. It can be seen that the proposed algorithm achieved a shorter makespan than the other algorithms.

To further verify the effect of the proposed algorithm, ten examples of Brandimarte dataset were tested with the results shown in Table III below.

<table>
<thead>
<tr>
<th>Example</th>
<th>n × m</th>
<th>S_{best}</th>
<th>Proposed algorithm</th>
<th>T_1</th>
<th>T_2</th>
<th>T_3</th>
<th>T_4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tx</td>
<td>Cmr</td>
<td>Tx</td>
<td>Cmr</td>
</tr>
<tr>
<td>MK01</td>
<td>10×6</td>
<td>25</td>
<td>27</td>
<td>30</td>
<td>5.1</td>
<td>30</td>
<td>5.1</td>
</tr>
<tr>
<td>MK02</td>
<td>10×6</td>
<td>22</td>
<td>22</td>
<td>18</td>
<td>6.7</td>
<td>18</td>
<td>6.7</td>
</tr>
<tr>
<td>MK03</td>
<td>15×8</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>0</td>
<td>168</td>
<td>0</td>
</tr>
<tr>
<td>MK04</td>
<td>15×8</td>
<td>32</td>
<td>36</td>
<td>34</td>
<td>9.7</td>
<td>30</td>
<td>8.5</td>
</tr>
<tr>
<td>MK05</td>
<td>15×4</td>
<td>124</td>
<td>124</td>
<td>128</td>
<td>6.2</td>
<td>128</td>
<td>6.2</td>
</tr>
<tr>
<td>MK06</td>
<td>10×12</td>
<td>37</td>
<td>49</td>
<td>56</td>
<td>15</td>
<td>58</td>
<td>16.7</td>
</tr>
<tr>
<td>MK07</td>
<td>20×5</td>
<td>117</td>
<td>119</td>
<td>125</td>
<td>4.2</td>
<td>125</td>
<td>4.2</td>
</tr>
<tr>
<td>MK08</td>
<td>20×10</td>
<td>326</td>
<td>326</td>
<td>326</td>
<td>0</td>
<td>326</td>
<td>0</td>
</tr>
<tr>
<td>MK09</td>
<td>20×10</td>
<td>278</td>
<td>285</td>
<td>292</td>
<td>3.2</td>
<td>288</td>
<td>2.6</td>
</tr>
<tr>
<td>MK10</td>
<td>20×15</td>
<td>166</td>
<td>170</td>
<td>191</td>
<td>18.3</td>
<td>182</td>
<td>12.7</td>
</tr>
<tr>
<td>MK11</td>
<td>20×20</td>
<td>191</td>
<td>163</td>
<td>175</td>
<td>19.1</td>
<td>191</td>
<td>13.2</td>
</tr>
<tr>
<td>MK12</td>
<td>30×30</td>
<td>256</td>
<td>190</td>
<td>192</td>
<td>22.9</td>
<td>220</td>
<td>19.8</td>
</tr>
</tbody>
</table>

As shown in Table III, the solutions obtained by the proposed algorithm are equal to or superior than those by other algorithms, except the example MK04.

B. SIMULATION

This subsection simulates a multi-objective FJSP of 6 jobs and 6 machines (Table IV), aiming to verify the effect of the proposed algorithm in handling dynamic FJSP.
For the FJSP in Table 4, if there is an urgent insertion of new job at time 5, the original plan must be adjusted. In this case, the Gantt charts before and after the adjustment are presented in Figures 2 and 3, respectively.

From Figure 3, it can be seen that the new job was prioritized over the jobs in the original plan. To maintain scheduling stability, the difference between the maximum makespan of the original job and the total makespan after insertion of the new job was minimized, and the machines of some remaining processes were also adjusted.

### VI. CONCLUSIONS

In actual production, the influencing factors of job-shop scheduling may change dynamically, leading to fuzziness in scheduling. Thus, this paper defines and classifies the dynamic multi-objective FJSP, and constructs a mathematical model according to the features of dynamic multi-objective FJSP. To acquire the optimal solution of dynamic FJSP, the author proposed the hybrid QPSO algorithm based on double chain coding. Through tests on Kacem and Brandimarte datasets, it is verified that the proposed strategy can quickly form an adjusted plan that has minimal difference from the original plan.

<table>
<thead>
<tr>
<th>Job</th>
<th>Process</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>O11</td>
<td>M1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>O12</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>O13</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>O14</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>O15</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>O16</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>O17</td>
<td>--</td>
</tr>
<tr>
<td>W2</td>
<td>O21</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>O22</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>O23</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>O24</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>O25</td>
<td>32</td>
</tr>
<tr>
<td>W3</td>
<td>O31</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>O32</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>O33</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>O34</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>O35</td>
<td>--</td>
</tr>
<tr>
<td>W4</td>
<td>O41</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>O42</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>O43</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>O44</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>O45</td>
<td>9</td>
</tr>
<tr>
<td>W6</td>
<td>O51</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>O52</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>O53</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>O54</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>O55</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>O56</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>O57</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>O58</td>
<td>15</td>
</tr>
</tbody>
</table>
REFERENCES


Wei Chen, male, born in Oct. 1983, is a professor at School of Management, Chongqing Technology and Business University. He received PHD in Management from Chongqing University in 2011. His major research interests include logistics and supply chain management, knowledge management and other related fields. As the project leader, he has undertaken two national social science fund projects of China, a key project of Chongqing Municipal Education Commission Humanities and Social Sciences, a social science project of Chongqing and other projects more than 10 items. As the first author or correspondent author, he has published more than 50 papers in international and domestic important academic journals, including Chinese Management Studies (SSCI), Education Science-Theory & Practice (SSCI), International Journal of Simulation Modelling (SCI), International Journal of u- and e-Service, Science and Technology (EI), Scientific Research Management (in Chinese). Besides, he has published four academic monographs in the Chinese social science press, science press, economic management press and other national publishing houses. He acquired the prize of outstanding doctoral thesis of Chongqing, the third prize of the 7th Institute of Higher Education Scientific Research Outstanding Achievements (Humanities and Social Sciences) (ranked second), the Excellent Paper Award of China Logistics Society and other more than ten research awards.

Hong Yang, female, born in March, 1977, is a professor at School of Geography and Tourism, Chongqing Normal University. She got PHD in Management from Chongqing University in 2009. Her major research interests include tourism management and planning, regional management and other related fields. As the project holder, she implemented over 10 projects, such as two national projects funded by National Office for Philosophy and Social Sciences. As the first or correspondent author, she published more than 20 papers in international and domestic important academic journals like Management World, Tourism Tribune, International Journal of Earth Sciences and Engineering, etc. Besides, she published 4 academic monographs.

Yifei Hao, male, born in Sep. 1983, is an associate professor at College of Mathematics and Statistics, Chongqing Technology and Business University. He received PhD in Mathematics from Lanzhou University in 2010. His major research interests include operation research, combinatorial optimization, supply chain management and other related fields. As the project leader, he has undertaken a Project of Chongqing Natural Science Foundation, a Humanity and Social Science Youth Foundation of Ministry of Education of China. As the first author or correspondent author, he has published several papers in international and domestic important academic journals, including Ars combinatorial (SCI), Semigroup Forum (SCI), Communications in Algebra (SCI), International Journal of Simulation Modelling (SCI), Technoeconomic & Management Research (in Chinese).