Research Article

Production Task Allocation Decision Based on Cloud Robot Cell-Line

Yi Du, Yongqiang Wang, Jialin Wang, and Tianpeng Zhang

1Anyang Institute of Technology School of Electronic Information and Electrical Engineering, Anyang 455000, China
2Anyang Branch, China Mobile Tietong, Anyang 455000, China
3Hangzhou Normal University, Hangzhou 310000, China

Correspondence should be addressed to Yi Du; 20170020@ayit.edu.cn

Received 11 January 2022; Revised 14 February 2022; Accepted 16 February 2022; Published 24 March 2022

Academic Editor: Daqing Gong

Copyright © 2022 Yi Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud manufacturing is a new service-oriented efficient and low-consumption agile manufacturing mode integrating information, manufacturing, Internet of Things, and other technologies. One of the key decisions of production enterprises is production task allocation based on cloud robot cell-line, which determines the efficiency and flexibility of the production system and affects various production links, such as job-shop logistics, production planning, and production scheduling. This paper explores the production task allocation, from the angle of the optimal combination of cloud manufacturing resources. First, a mathematical model was established, based on the transport cost of different sub-tasks and the tardiness cost of product delivery, and solved by quantum firefly algorithm (QFA). Next, QFA was proved superior to traditional firefly algorithm (FA), improved FA, and the FA optimized by cat swarm optimization (CSO-FA), in terms of time complexity and spatial complexity. The research enriches the theory and methodology of allocating operation-level cloud manufacturing resources based on cloud robot cell-line and provides decision support to manufacturers, which want to implement operation-level allocation of cloud manufacturing resources based on cloud robot cell-line.

1. Introduction

Big manufacturers like China are improving the intelligence level of the manufacturing industry. Cloud intelligence can increase the utilization efficiency of production lines, lower the production cost, and avoid repeated investment and construction, contributing to the improvement of our living standards.

Unlike traditional flow lines, the cell-line combines a few similar stations into a novel type of production line. There are two advantages of the cell-line: First, it is of the same production efficiency as a conveyor belt flow line. Second, the flexible production of production units adapts well to the market needs of various small batches of products. Therefore, the cell-line is regarded as an ideal production mode to realize massive customization. Besides, the cell-line provides a green and economic production method.

Scholars at home and abroad have explored robotic automation production lines extensively. Focusing on the variable process pathway, Tian et al. [1] established a minimum delay time function and constructed a scheduling module, in the light of collaborative, differentiated agents. In addition, agents were adopted to realize global optimization, aiming to minimize the makespan. Zacharia and Nearchou [2] summarized the features of job-shop scheduling with multiple robotic manufacturing units, created a mathematical model for minimizing the makespan of the problem, and solved the model effectively with a hybrid genetic algorithm. Liu et al. [3] utilized the minimum completion time to solve the flexible job-shop scheduling with loading/unloading robots, established a mathematical model of the problem, and searched for the scheduling results with a tabu search and greedy algorithm. Miyake [4] used the minimum production cycle to solve the multi-process job-shop scheduling with transport time, constructed a nonlinear
mathematical model, and solved the model with improved genetic algorithm, realizing batch scheduling. Du et al. [5] explored the reconstruction and scheduling of multi-stage variable-batch production lines in labor-intensive enterprises, adopted different algorithms for production lines of different scales, and compared the applicable scopes of two joint optimization algorithms with examples. Wu et al. [6] studied the optimal control of event-driven conveyor belt feeding and processing system.

There are some limitations that present opportunities for future research. Our Stackelberg game modeled the operator as the leader and the suppliers as the followers. It might be beneficial for practical applications to explore other power structures. In future research, a supplier-led power structure (e.g., a large-scale enterprise building a platform) could be an important extension of this model. Moreover, we studied two types of substitutable cloud services that can satisfy client demands. In reality, there might be multiple types of cloud services that meet the same requirement. Future research should consider the case of multiple types of substitutable cloud services in the CMfg system.

Cao et al. [7] stated that there are some limitations that present opportunities for future research. Our Stackelberg game modeled the operator as the leader and the suppliers as the followers. It might be beneficial for practical applications to explore other power structures. In future research, a supplier-led power structure (e.g., a large-scale enterprise building a platform) could be an important extension of this model. Moreover, we studied two types of substitutable cloud services that can satisfy client demands. In reality, there might be multiple types of cloud services that meet the same requirement. Future research should consider the case of multiple types of substitutable cloud services in the CMfg system. Carlucci et al. [8] observed that a key limitation of the research is the use of homogeneous resources shared in the network. Future research could take into account different categories of resources that are characterized by different rules in order to determine the win or lose state. Another important research issue to further investigate is the dynamic composition of the network based on the obtained performance, i.e., how the participant in the network can decide to enter or leave the network.

To sum up, there are very few studies on cell-line cloud manufacturing resource scheduling based on cloud robots, which could avoid resource waste, repetitive building of production lines, and low utilization of units. This paper explores production task allocation from the angle of the optimal combination of cloud manufacturing resources. Specifically, a mathematical model was established based on the transport cost of different sub-tasks and the tardiness cost of product delivery, and solved by quantum firefly algorithm (QFA). The effectiveness of the model and algorithm was demonstrated through example analysis.

At present, there are many researches on the single technology of cloud manufacturing. With the in-depth research and application of cloud manufacturing, the cloud manufacturing service platform oriented to the whole life cycle of products will become the next research hotspot. New product development is an important life cycle stage of complex electronic products. There are very few researches on cloud manufacturing resource scheduling of the Japanese unit production line based on cloud robot. However, the results of researches can avoid the waste of resources, repeated line construction, and low utilization rate of units.

The driving effect of electronic information industry to China’s economic growth is showing a sustained and scale of growth. Moreover, the promotion and penetration effect to the upgrading of traditional industrial structure gradually increases. The high-tech industry represented by the electronic information industry has developed far faster than the traditional industry, and the export of high-tech products has shown a rapid growth trend. By observing the development of electronic information industry and the track of technological upgrading, it is concluded that China has a strong and obvious comparative advantage in processing and assembly, and the manufacturing of electronic information products in the world is forming a tendency to be concentrated in some areas of China. First, due to the epidemic, foreign orders have increased, and some manufacturers have blindly expanded production scale, resulting in serious excess production capacity. However, some manufacturers have insufficient orders and idle production resources. Second, the processing technology of small batch products is more and more complex and often the production resources of an enterprise cannot meet the requirements of product processing. The cloud platform can contact enterprises in multiple regions to complete manufacturing resource scheduling tasks together.

2. Problem Formulation

In 2010, James first put forward the definition of “cloud” robot; it points out that the cloud robotics is robot to store and retrieve information, such as a robot with a camera around the photo upload cloud server, a search on the server, find the similar photos, to plan the robot path tracing, and store the pictures, as search information for the back of the machine. All robots can share this library to reduce the developer effort and time.

Suppose there are requests for $I$ types of products on the cloud platform. Each product request contains $N$ sub-tasks $F_i$, $i = \{1, 2, \ldots, N\}$ to be processed on $k$ intelligent cloud scheduling production lines. Each sub-task contains $P_i$ operations, and $K_i$ optional production lines. The production task can be expressed as follows:

$$F = \min_{(f_1, f_2)}$$

where $f_1$ is the minimum processing time of the manufacturing resources of a task, which depends on the sub-task with the longest processing time:

$$f_1 = \min \left( \max \sum_{i=1}^{I} \sum_{k=1}^{K} \left( c_{i,k} x_{i,k} + f_{i,k} h_{i,k} \right) \right)$$

where $c_{i,k}$ is a $0$–$1$ variable (if $c_{i,k} = 1$, product $i$ is processed through $t$ stages on production line $k$; if $c_{i,k} = 0$, product $i$ is not processed through $t$ stages on production line $k$); $f_{i,k}$ is
the number of product $i$ through $t$ stages; $l_{ikt}$ is the processing time of product $i$ through $t$ stages on production line $k$; $q_{ik}$ is the transport time of 1,000 products $i$ from production line $k$ to production line $k+1$; $x_{ikt+i}$ is a 0–1 variable (if $x_{ikt+i} = 1$, product $i$ is transported from production line $k$ to production line $k+1$ through $t$ stages; if $x_{ikt+i} = 0$, product $i$ is not transported from production line $k$ to production line $k+1$ through $t$ stages); $h_{it}$ is the multiple of 1,000 products $i$ through $t$ stages.

$f_2$ is the minimum cost of all sub-tasks of manufacturing resource processing:

$$ f_2 = \min \left( \max \sum_{t=1}^{M_i} \sum_{k=1}^{K} \left( e_{ilk} \epsilon_{it} C_{itk} + (f_{itk} - s_{itk}) * d_{it} \right) \right) ,$$

where $C_{itk}$ is the manufacturing cost of product $i$ through $t$ stages on production line $k$; $f_{itk}$ is the processing time of product $i$ through $t$ stages on production line $k$ (including transport time); $s_{itk}$ is the delivery period of product $i$ through $t$ stages on production line $k$; and $d_{itk}$ is the tardiness cost constraint of product $i$ through $t$ stages.

The processing of the current operation of a sub-task should not begin before the completion of the previous operation:

$$ \sum_{k=1}^{M_i} e_{ijkt} = 1 ,$$

where $t_{end,j}$ and $t_{start,(i+j)}$ are the completion time of the previous operation and the start time of the current operation, respectively.

Operation is the minimum granularity of scheduling. Each operation can only occupy one logic manufacturing resource.

### 3. Algorithm Design

The QFA is designed in the following steps [9–14]. First, a swarm of $h$ quantum fireflies is generated. Each quantum firefly has a quantum position and a position. The position of each quantum firefly is represented by a D-dimensional numerical string [0, 1], with $D$ being the dimensionality of the solution space. In the $t$-th iteration, the quantum position of the i-th quantum firefly can be expressed as follows:

$$ \psi_{il}^t = \left[ \begin{array}{c} \alpha_{i1}^t \\ \alpha_{i2}^t \\ \vdots \\ \alpha_{iD}^t \\ \beta_{i1}^t \\ \beta_{i2}^t \\ \vdots \\ \beta_{iD}^t \end{array} \right] ,$$

where $(\alpha_{il}^t)^2 + (\beta_{il}^t)^2 = 1$ (i = 1, 2, ..., $D$). The quantum positions $\alpha_{il}^t$ and $\beta_{il}^t$ satisfy $0 \leq \alpha_{il}^t \leq 1$, and $0 \leq \beta_{il}^t \leq 1$. At the beginning, all the quantum positions of the quantum fireflies are initialized as 1. For simplicity, the quantum position of the i-th quantum firefly in the $t$-th iteration can be denoted as $\psi_{il}^t = (\psi_{il1}^t, \psi_{il2}^t, ..., \psi_{ilD}^t)$ ($i = 1, 2, ..., h$). The position of a quantum firefly can be derived from the measured quantum position. The position of the i-th quantum firefly can be expressed as $x_{il}^t = (x_{i1l}^t, x_{i2l}^t, ..., x_{iDl}^t)$ ($i = 1, 2, ..., h$). The global optimal position of the i-th quantum firefly in the t-th iteration can be expressed as $P_{gl}^t = (P_{g1l}^t, P_{g2l}^t, ..., P_{gDl}^t)$.

According to the fluorescein update rule, the fitness $F$ ($p_{it}^t$) of quantum firefly $i$ ($i = 1, 2, ..., h$) at the local optimal position in the $t$-th iteration can be converted into fluorescein $L_i(t)$:

$$ L_i(t) = (1 - \gamma)L_i(t - 1) + eF(p_{it}^t) ,$$

where $\gamma \in [0, 1]$ is the disappearance rate of fluorescein, which gradually weakens with the growing distance and media absorption; $e$ is the update rate of fluorescein. The i-th quantum firefly obtains the learning neighborhood by certain rules. The quantum fireflies are selected from the neighborhood based on the fluorescein value and position similarity. In the $t$-th iteration, the i-th quantum firefly obtains the learning neighborhood by the following equation:

$$ N_i(t) = \{ q|D - d_{iq}(t) \leq r_1(t) \bar{L}_i(t) \leq L_q(t) \} ,$$

where $N_i(t)$ is the set of labels of the learning neighborhood of the i-th quantum firefly; $d_{iq}(t)$ is the distance between the local optimal positions of the i-th and q-th quantum fireflies; $r_1(t)$ is the radius of the dynamic decision domain of the i-th quantum firefly; and $L_q(t)$ is the fluorescein value of the q-th quantum firefly in the $t$-th iteration. The number of quantum fireflies in the learning neighborhood equals the number of labels in the label set of that neighborhood in the current iteration.

In each iteration, each quantum firefly selects its moving direction according to the quantum fireflies in the neighborhood. In the $t$-th iteration, the probability for the i-th quantum firefly to move to the q-th quantum firefly in its neighborhood can be expressed as

$$ P_{iq}^t = \frac{L_q(t) - L_i(t)}{\sum_{j=1}^{N_i(t)} L_j(t) - L_i(t)} .$$

In each iteration, if the learning neighborhood of the i-th quantum firefly is empty, then the evolution of the i-th dimensional quantum position of that quantum firefly can be expressed as follows:

$$ \psi_{il}^{t+1} = \begin{cases} N \psi_{il}^t \theta_{il}^{t+1} = 0 \text{ and } d_{il}^{t+1} < c_1 \\ \{ \text{abs} \left( U(\theta_{il}^{t+1}) \psi_{il}^t \right) \} \end{cases} ,$$

where quantum rotation angle $\theta_{il}^{t+1} = \theta_{il}^1 (p_{gl} - x_{il}^t) + \theta_{il}^2 (p_{gl}^t - x_{il}^t)$ ($i = 1, 2, ..., h$; $\theta_{il}^1 = 1, 2, ..., D$; $\epsilon_1$ and $\epsilon_2$ are the degree of influence of local optimal position and global optimal position on quantum rotation angle, respectively; $\mu_{il}^{t+1}$ is a random number uniformly distributed in [0, 1]; $c_1$, a constant in [0, 1/$D$], is the mutation probability of the quantum position at the quantum rotation angle of $\theta$; abs(.) is the absolute value that limits each dimension of the quantum position in [0, 1]; $U(\theta_{il}^{t+1}) = \begin{bmatrix} \cos \theta_{il}^{t+1} & -\sin \theta_{il}^{t+1} \\ \sin \theta_{il}^{t+1} & \cos \theta_{il}^{t+1} \end{bmatrix}$ is the quantum rotation gate; and $\bar{N} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is the quantum NOT gate.
If the learning neighborhood of the i-th quantum firefly is not empty, each quantum firefly labeled z in the neighborhood satisfies \( z = \arg \max_{\mathbf{N}_i} \mathbb{P}_i(0) \). In the i-th iteration, then the evolution of the l-th dimensional quantum position of the i-th quantum firefly can be expressed as follows:

\[
\begin{aligned}
\eta_{il}^{t+1} = & \begin{cases}
\mathbb{N}_il_{u} & \theta_{il}^{t+1} = 0 \text{ on } d \mu_{il}^{t+1} < c_2 \\
\abs{U(\theta_{il}^{t+1})} & ,
\end{cases}
\end{aligned}
\]

(10)

where quantum rotation angle \( \theta_{il}^{t+1} = e_5 (p_{il}^l - x_{il}^l) + e_4 (p_{il}^{l-1} - x_{il}^{l-1}) + e_5 (p_{il}^{l-2} - x_{il}^{l-2}) \); \( l = 1, 2, ..., h; h = 1, 2, ..., D; p_{il}^l \) is the l-th dimension of the local optimal position, i.e., the position with the highest fluorescein, in the learning neighborhood of the i-th quantum firefly; \( e_3, e_4, \) and \( e_5 \) are the degree of influence of the local optimal position of the i-th quantum firefly, the local optimal position, i.e., the position with the highest fluorescein, in the learning neighborhood of the i-th quantum firefly, and the global optimal position of the i-th quantum firefly on quantum rotation angle, respectively; \( c_3 \), a constant in \([0, 1/D]\), is the mutation probability of the quantum position at the quantum rotation angle of 0. The position of the i-th quantum firefly can be derived from the measured quantum position:

\[
x_{il}^{t+1} = \begin{cases}
1, & \eta_{il}^{t+1} > (a_{il}^{t+1})^2 \\
0, & \eta_{il}^{t+1} \leq (a_{il}^{t+1})^2
\end{cases},
\]

(11)

where \( l = 1, 2, ..., D; \eta_{il}^{t+1} \in [0, 1] \) is a random number obeying uniform distribution; \( (a_{il}^{t+1})^2 \) is the probability of zero appearing at quantum position \( v_{il}^{t+1} \).

The radius of the dynamic decision domain of the i-th quantum firefly can be updated by the following expression:

\[
r_i(t + 1) = \min[R_i, \max[0, r_i(t) + \xi[n_i - \text{size}[N_i(t)]]]] ,
\]

(12)

where \( \xi \) is a constant update rate of the dynamic decision domain; \( R \geq r_i(t) \) is a constant perception domain; \( \text{min} \) and \( \text{max} \) are the minimum and maximum functions, respectively; \( n_i \) is the parameter controlling the number of quantum fireflies in the learning neighborhood; \( \text{size}[N_i(t)] \) is the number of quantum fireflies in the learning neighborhood of the i-th quantum firefly.

For the cloud scheduling of production line, the fitness function of the QFA should minimize the production time of the production line. After computing the maximum production time and cost of cloud scheduling, the minimization of the two values is adopted as the optimization goal. Then, the fitness function of the current position \( x_i^l = (x_{i1}^l, x_{i2}^l, \ldots, x_{id}^l) (i = 1, 2, \ldots, h) \) of the i-th firefly can be expressed as follows:

\[
F(x_i^l) = \begin{cases}
-\text{MSLL}(x_i^l), & \text{IF crat} \leq \text{erat} \\
-\rho \cdot \text{MSLL}(x_i^l), & \text{otherwise}
\end{cases},
\]

(13)

where \( \text{MSLL}(x_i^l) \) is the maximum production time and cost of cloud scheduling constructed by \( x_i^l; \rho \leq 1; \text{crat} \) and \( \text{erat} \) are the calculated and expected coverage rates, respectively [15–19].

Based on the QFA, the production time and cost of the cloud manufacturing production line can be minimized through the following steps:

1. Establish the mathematical model of the production time and cost of cloud computing production line, determine the key parameters of the QFA corresponding to the production time and cost, initialize the quantum positions of quantum fireflies, and measure their positions.

2. Map the position of each quantum firefly to the corresponding parameters of the established model, import these parameters to the fitness function, compute the fitness at the position of the quantum firefly, and determine the local and global optimal positions according to the fitness.

3. Update the fluorescein value and learning neighborhood of each quantum firefly, according to the fitness at the local optimal position of that quantum firefly.

4. Change the quantum position of each quantum firefly with new data.

5. Change the radius of the dynamic decision domain of each quantum firefly with new data.

6. Compute the fitness of each quantum firefly at the new position by the fitness function and determine the local and global optimal positions according to the fitness again.

7. If the maximum number of iterations is reached, execute Step 8; otherwise, return to Step 3.

8. Output the global optimal position [20–25]. (Flow chart as Figure 1)

The test example and calculation results are shown in Tables 1 and 2, respectively.

4. Example Analysis

Two swarms were initialized in our example. For convenience, each type of the product can be used after completion one production cycle. The initial swarms 1 and 2 were combinations of ones in each row of Table 1. Our algorithm IFA was compared with traditional firefly algorithm (FA), improved FA, and the FA optimized by cat swarm optimization (CSO-FA). The swarm size for all these intelligent optimization algorithms was fixed at 50 (Set in the program). The maximum number of iterations was set to 350 (Set in the program), the expected coverage rate to 0.7 (In Formula (13), \( \text{erat} \) is the expected filling rate), \( \rho = 0.001 \) (In Formula (13) \( \rho \leq 1 \)), the update rate of dynamic decision domain to 0.8 (Formula (12) \( \xi \) is the update rate of the dynamic decision domain), the initial fluorescein value to 5, \( e_1 = 0.06 \), \( e_2 = 0.03 \), \( e_3 = 0.06 \), \( e_4 = 0.03 \), \( e_5 = 0.01 \), \( \theta_1^{t+1} = e_4 (p_{il}^l - x_{il}^l) + e_5 (p_{il}^{l-1} - x_{il}^{l-1}) + e_5 (p_{il}^{l-2} - x_{il}^{l-2}) \); \( i = 1, 2, ..., h; h = 1, 2, ..., D; \rho_{il} \) and \( e_2 \) are the degree of influence of local optimal position.
Establish the mathematical model of the production time and cost of cloud computing production line, determine the key parameters of the QFA corresponding to the production time and cost, initialize the quantum positions of quantum fireflies, and measure their positions.

Map the position of each quantum firefly to the corresponding parameters of the established model, import these parameters to the fitness function, compute the fitness at the position of the quantum firefly, and determine the local and global optimal positions according to the fitness.

Update the fluorescein value and learning neighborhood of each quantum firefly, according to the fitness at the local optimal position of that quantum firefly.

If the maximum number of iterations is reached, output the global optimal position.

begin

end

Figure 1: Program flow chart.

Table 1: Example of cloud manufacturing resource allocation.

<table>
<thead>
<tr>
<th>Order</th>
<th>( d_i ) (10,000 yuan/1,000 pieces/day)</th>
<th>Suborder</th>
<th>Production line</th>
<th>( c_{ik} ) (10,000 yuan/1,000 pieces)</th>
<th>( l_i )/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J_1 )</td>
<td>1</td>
<td>( F_{1,1} )</td>
<td>( K_3/K_5/K_7 )</td>
<td>3/3/5</td>
<td>5/5/3</td>
</tr>
<tr>
<td>( J_2 )</td>
<td>1</td>
<td>( F_{2,1} )</td>
<td>( K_7/K_5/K_3 )</td>
<td>5/8/8</td>
<td>8/5/5</td>
</tr>
<tr>
<td>( J_3 )</td>
<td>1</td>
<td>( F_{3,1} )</td>
<td>( K_1/K_5/K_10 )</td>
<td>6/3</td>
<td>3/6</td>
</tr>
<tr>
<td>( J_4 )</td>
<td>1</td>
<td>( F_{4,1} )</td>
<td>( K_3/K_4/K_7 )</td>
<td>4/3</td>
<td>3/4</td>
</tr>
<tr>
<td>( J_5 )</td>
<td>1</td>
<td>( F_{5,1} )</td>
<td>( K_5/K_6/K_9 )</td>
<td>3/6/4</td>
<td>6/3/4</td>
</tr>
<tr>
<td>( J_6 )</td>
<td>1</td>
<td>( F_{6,1} )</td>
<td>( K_2/K_6 )</td>
<td>2/6/2</td>
<td>6/2/6</td>
</tr>
<tr>
<td>( J_7 )</td>
<td>1</td>
<td>( F_{7,1} )</td>
<td>( K_3/K_4 )</td>
<td>3/7</td>
<td>7/3</td>
</tr>
<tr>
<td>( J_8 )</td>
<td>1</td>
<td>( F_{8,1} )</td>
<td>( K_2/K_6/K_9 )</td>
<td>5/4/3</td>
<td>3/4/5</td>
</tr>
<tr>
<td>( J_{9} )</td>
<td>1</td>
<td>( F_{9,1} )</td>
<td>( K_4/K_7/K_10 )</td>
<td>4/3/4/6</td>
<td>4/6/4/3</td>
</tr>
</tbody>
</table>
and global optimal position on quantum rotation angle, $\hat{\theta}_i^{l+1} = c_1 (p_{il}^l - x_i^l) + c_2 (p_{il}^{l+1} - x_i^l) + c_3 (p_{il}^l - x_i^l)$; $i = 1, 2, ..., h; l = 1, 2, ..., D; p_{il}^l$ is the $l$-th dimension of the local optimal position, i.e., the position with the highest fluorescein, in the learning neighborhood of the $i$-th quantum firefly; $c_1$, $c_2$, and $c_3$ are the degree of influence of the local optimal position of the $i$-th quantum firefly) and $c_1 = c_2 = 0.1 / D$.

5. Conclusions

This paper mainly solves the cloud robot scheduling of a cell-line involving various small batches of products, which are produced in multiple cycles, in the light of transport cost. The QFA was adopted to reasonably allocate the resources in the cloud environment. Besides, the ideas of quantum computing were added to the FA to speed up the search, while maintaining the swarm search ability. Compared with the traditional FA, improved FA, and CSO-FA, the proposed QFA (as Figures 2 and 3) achieves excellent optimization ability and a fast search speed. Finally, our algorithm was demonstrated through an example analysis on the production order scheduling of an electronic product manufacturer. The results show that the QFA is capable of effectively solving the multiobjective scheduling of cloud manufacturing resources [26, 27].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

<table>
<thead>
<tr>
<th>Table 1: Continued.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$J_7$</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$J_8$</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Table 2: Transport cost.

<table>
<thead>
<tr>
<th>Production line transport cost (10,000 yuan/1,000 pieces)/(day/1,000 pieces)</th>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$K_3$</th>
<th>$K_4$</th>
<th>$K_5$</th>
<th>$K_6$</th>
<th>$K_7$</th>
<th>$K_8$</th>
<th>$K_9$</th>
<th>$K_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>0/0</td>
<td>2/1</td>
<td>1/0/5</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
<td>4/2</td>
<td>4/2</td>
<td>1/0/5</td>
<td>2/1</td>
</tr>
<tr>
<td>$K_2$</td>
<td>3/0/5</td>
<td>0/0</td>
<td>2/1</td>
<td>3/1/5</td>
<td>1/0/5</td>
<td>2/1</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
<td>1/0/5</td>
</tr>
<tr>
<td>$K_3$</td>
<td>2/0/5</td>
<td>3/1/5</td>
<td>0/0</td>
<td>1/0/5</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
</tr>
<tr>
<td>$K_4$</td>
<td>1/0/5</td>
<td>2/1</td>
<td>2/1</td>
<td>0/0</td>
<td>1/0/5</td>
<td>3/1/5</td>
<td>3/1/5</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
</tr>
<tr>
<td>$K_5$</td>
<td>1/0/5</td>
<td>3/1/5</td>
<td>1/0/5</td>
<td>3/1/5</td>
<td>0/0</td>
<td>2/1</td>
<td>4/2</td>
<td>3/1/5</td>
<td>2/1</td>
<td>3/1/5</td>
</tr>
<tr>
<td>$K_6$</td>
<td>2/0/5</td>
<td>2/1</td>
<td>2/1</td>
<td>4/2</td>
<td>2/1</td>
<td>0/0</td>
<td>3/1/5</td>
<td>3/1/5</td>
<td>4/2</td>
<td>2/1</td>
</tr>
<tr>
<td>$K_7$</td>
<td>3/0/5</td>
<td>3/1/5</td>
<td>3/1/5</td>
<td>3/1/5</td>
<td>2/1</td>
<td>3/1/5</td>
<td>0/0</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
</tr>
<tr>
<td>$K_8$</td>
<td>2/0/5</td>
<td>3/1/5</td>
<td>3/1/5</td>
<td>2/1</td>
<td>2/1</td>
<td>2/1</td>
<td>2/1</td>
<td>0/0</td>
<td>2/1</td>
<td>3/1/5</td>
</tr>
<tr>
<td>$K_9$</td>
<td>3/0/5</td>
<td>2/1</td>
<td>4/2</td>
<td>2/1</td>
<td>3/1/5</td>
<td>4/2</td>
<td>3/1/5</td>
<td>1/0/5</td>
<td>0/0</td>
<td>3/1/5</td>
</tr>
<tr>
<td>$K_{10}$</td>
<td>1/0/5</td>
<td>2/1</td>
<td>2/1</td>
<td>3/1/5</td>
<td>2/1</td>
<td>5/2/5</td>
<td>2/1</td>
<td>3/1/5</td>
<td>3/1/5</td>
<td>0/0</td>
</tr>
</tbody>
</table>

Figure 2: Convergence curves of different algorithms.

Figure 3: Operation time of different algorithms.
Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Anyang Key Science and Technology Project (anke[2020]14–48; 2021A01SF007), Key Science and Technology Project, Anyang Institute of Technology, and Key Scientific Research Projects of Universities (YPY2020009), and Henan Provincial Department of Education (20B460002).

References


