SIMULATION STUDY ON MULTI-OBJECTIVE BLOCKING LOT STREAMING FLOW SHOP SCHEDULING BASED ON IMPROVED ARTIFICIAL BEE COLONY ALGORITHM

Gaizhen YANG
Huanggang Normal University Department of Computer Science, Huanggang 438000, China
E-mail: jyjygz@hgnu.edu.cn

ABSTRACT: In the flexible flow shop scheduling of actual production, there often exist the factors such as blocking of product processing and uncertain processing time, so it’s of great practical significance to study the optimization of multi-objective flow shop scheduling under the blocking conditions. For this, the paper establishes an optimization model of multi-objective flow shop scheduling considering plugging effect. Then, the artificial bee colony (ABC) algorithm was improved to be used in the model calculation. Finally, the improved artificial bee colony (IABC) algorithm proposed in this paper was verified by simulation examples. Specifically, in the algorithm optimization phase, the heuristic algorithm MME and NEH were first used to generate the initial population. After generating the new excellent global solution based on the SBOX algorithm in the hired bee phase, the PLS algorithm was adopted to generate more neighborhood solutions, which can significantly improve the computational convergence speed of the ABC algorithm and help the initial population to evolve more efficiently toward the real Pareto frontier. The simulation test results show that the proposed algorithm can obtain higher-quality non-dominated solutions under the same problem scale and calculation time, and with the number of iterations increasing, the utilization degree of non-dominated solutions and the evolution rate of populations, as well as the scale of high-quality solutions gradually increase. The research findings can provide a theoretical reference for the workshop optimization of multi-objective scheduling.

KEYWORDS: production workshop, lot-streaming flow shop scheduling, multi-objective optimization, blocking, improved artificial bee colony algorithm (IABC)

1 INTRODUCTION
Lot-streaming flow shop scheduling is the main scheduling mode of production workshops in recent years. It splits the same type of products into several sub-lots according to their structural composition, and then the sub-lots can be transported to the subsequent machine for continuous processing after completing the processing in the upstream (Filho and Bahiense, 2013; Mirhassani and Ghorbanalizadeh, 2008), which significantly reduces total product processing time and machine idle time. Thus, this scheduling method has been widely used in various industries such as machinery, textile, automotive, and chemical industries (Hill et al., 2016; Mostafaei et al., 2015:).

At present, single-objective flow shop scheduling and simple multi-objective flow shop scheduling methods in production workshops are relatively mature. In single-objective flow shop scheduling, researchers mainly use genetic algorithm, local search algorithm, particle swarm algorithm, ant colony algorithm and other methods to solve the lot-streaming flow of products (Defersha, 2012; Fattahi et al, 2013; Govardhan and Roy, 2015; Huang and Yin, 2003; Li, 2012; Liu and Liu, 2008; Martin, 2009; Tseng and Liao, 2008; Tseng and Lin, 2009; Yang and Wang, 2012:). Multi-objective flow shop scheduling is an extension of single-objective scheduling, also taking into account optimization objectives such as processing cost, processing lot, completion time, and delay time etc. However, due to constraints or even contradictions between the above optimization objectives, most researchers have converted the multi-objective optimization function to the weighting of a single objective function (Baykasoglu et al., 2004; Chen, 2010; Xia and Wu, 2005; Zhang et al., 2009). For instance, Bandyopadhyay et al. used the product processing time, delay time and cost as the objective functions, to solve the established optimization model by NSGA-II algorithm (Bandyopadhyay and Bhattacharya, 2013); Li et al. studied the correspondence between product processing time and machine processing load, transforming the two according to the weight ratio, and used the improved emergency search algorithm to obtain the
optimal production scheduling scheme (Li et al., 2010; Li et al., 2011); Zhang et al. (2010) proposed
the concept of decision-level indicators and introduced it into the multi-objective flow shop
optimization scheduling, so that the decision-level indicators can be used to prioritize individuals
which have the same degree of importance in the scheduling process.

At present, there have been relatively few studies on production flow shop scheduling of
multiple optimization objectives. The existing research has the following problems: First, when
converting multi-objective functions into single-objective functions, the weights of each objective
function are determined mostly based on experience, lacking rigorous theoretical basis;
second, the non-directionality of the crossover and variation operation etc. in the traditional scheduling
model seriously reduces the search speed and convergence speed of the optimal solution; third, in
the actual production workshop scheduling, there are also many uncertainties such as machine
failures, new product orders and product processing blockages etc., but very few studies have been
conducted on the lot-streaming flow shop optimization scheduling problem based on both the
multi-objective functions and uncertainties blocking currently (Aghezzaf et al., 2010; Liu et al. 2015).

In view of the above defects, this paper establishes a multi-objective flow shop scheduling model of production workshop considering the clogging effect. Then, it improves the ABC algorithm, and uses it to solve the established model. Finally, the IABC algorithm proposed in this paper was verified using the simulation example. The research findings can provide a theoretical
reference for the multiple-objective optimization scheduling of production workshop.

2 MULTI-OBJECTIVE FLOW SHOP
SCHEDULING MODEL
CONSIDERING BLOCKING EFFECT

Let the processing sequence of a certain type of product be:
\[ \pi = \{ \pi_1, \pi_2, \pi_3, \ldots, \pi_j, \ldots, \pi_n \} \] (1)

All workpieces follow the same planned processing route. It’s assumed that the number of
machines is m, and any workpiece in the sequence is split into l sublots. Then,
\[ \pi_j = \{ \pi_j^1, \pi_j^2, \pi_j^3, \ldots, \pi_j^l \} \] (2)

The assumptions have been made about the processing of the production workshop as follows:
(1) Only after all the sublots of the same product has been processed can the next type of product be
processed;
(2) The processing machine \( m \) can only process one sublot at the same time, and one sublot must be
processed by the designated machine;
(3) No buffer is set between the adjacent two processing machines;
(4) Certain waiting time is set between two adjacent sublots processed on the same machine;
(5) The processing preparation time and product transportation time of the processing machine are
calculated within the overall processing time.

Taking two products as an example, the general lot-streaming flow shop scheduling and blocking
effect-based flow shop scheduling were discussed. Figure 1 shows the Gantt chart of the two.

![Fig. 1 Gantt chart of the general production workshop lot-streaming scheduling and blocking flow shop scheduling](image-url)
It can be seen from Figure 1 above that after M1 has machined π(1), since M2 processes π(1) for a long time and there is no buffer between M1 and M2, π(1) can only stay on M1, waiting for M2 to finish processing π(1) and then move to M2, and so on, π(1) can only be further post-processed. In addition, the maximum production processing time without blocking is 18, while the maximum production processing time with blocking is 19.

For this, a multi-objective flow shop scheduling model considering the blocking effect was established in this study, in which the objective function $f_1$ indicates the maximum completion time of product processing under blocking conditions; $f_2$ indicates the delayed completion time of product processing.

\[
\min f_1 = C_{x_{(s)\cdot u\cdot j_{(s)}}} \quad (3)
\]

\[
f_2 = \sum_{j=1}^{n} \max \left(0, d_j - C_{x_{j\cdot u\cdot j_{(s)}}} \right) \quad (4)
\]

In the formula above, $C_{m_{th}\cdot m\cdot l}$ is the completion time of the $l$-th sublot for the $n$th workpiece on the $m$-th processing machine; $d_i$ is the final delivery date of the $n$th workpiece.

Equation 5-12 shows the calculating process of $f_1$.

\[
\begin{align*}
S_{x_{(j)\cdot l\cdot l_{(j)}}} &= 0 \\
C_{x_{(j)\cdot l\cdot l_{(j)}}} &= S_{x_{(j)\cdot l\cdot l_{(j)}}} + p_{x_{j\cdot l_{(j)}}} \quad (5)
\end{align*}
\]

\[
\begin{align*}
S_{x_{(j)\cdot l\cdot l_{(j)}}} &= C_{x_{(j)\cdot l\cdot l_{(j)}}} \\
C_{x_{(j)\cdot l\cdot l_{(j)}}} &= S_{x_{(j)\cdot l\cdot l_{(j)}}} + p_{x_{j\cdot l_{(j)}}} \quad (6)
\end{align*}
\]

\[
\begin{align*}
S_{x_{(j)\cdot l\cdot l_{(j)}}} &= \max \left\{ C_{x_{(j)\cdot l\cdot l_{(j)}}} , S_{x_{(j)\cdot l\cdot l_{(j)}}} \right\} \\
C_{x_{(j)\cdot l\cdot l_{(j)}}} &= S_{x_{(j)\cdot l\cdot l_{(j)}}} + p_{x_{j\cdot l_{(j)}}} \quad (7)
\end{align*}
\]

\[
\begin{align*}
S_{x_{(j)\cdot l\cdot l_{(j)}}} &= \max \left\{ C_{x_{(j)\cdot l\cdot l_{(j)}}} , S_{x_{(j)\cdot l\cdot l_{(j)}}} \right\} \\
C_{x_{(j)\cdot l\cdot l_{(j)}}} &= S_{x_{(j)\cdot l\cdot l_{(j)}}} + p_{x_{j\cdot l_{(j)}}} \quad (8)
\end{align*}
\]

\[
\begin{align*}
S_{x_{(j)\cdot l\cdot l_{(j)}}} &= \max \left\{ C_{x_{(j)\cdot l\cdot l_{(j)}}} , S_{x_{(j)\cdot l\cdot l_{(j)}}} \right\} \\
C_{x_{(j)\cdot l\cdot l_{(j)}}} &= S_{x_{(j)\cdot l\cdot l_{(j)}}} + p_{x_{j\cdot l_{(j)}}} \quad (9)
\end{align*}
\]

where, $l_{e(j)}$ represents the number of sublots included in the product; $p_{n(j)}$ is the processing time of the product on the $t$-th processing machine; $S_{a(j)\cdot e\cdot e_{(j)}}$ is the processing starting time of the $e$-th sublot.

### 3 Improved Artificial Bee Colony Algorithm

#### 3.1 Initialization optimization of population

In this paper, the traditional ABC algorithm was improved, and then used in the model solution. The population of the ABC algorithm was initialized using the heuristic algorithms MME and NEH.

Firstly, the MME algorithm was used for population initialization optimization of $f_1$ in the basic steps as follows:

(a) Use the MME algorithm to generate a set of scheduling sequences $\pi$, select the two products $\pi(1)$, $\pi(n)$ with the least processing time from the sequence, respectively, and put them into the first and last machine for processing in the workshop;

(b) Form a new set $U$ with the remaining $\pi$-2 products, and select the product corresponding to the machine by the $F$ value.

\[
F_i = \theta \times \sum_{j=1}^{m-1} p_{x_{j\cdot l_{(j)}}} - p_{x_{j\cdot l_{(j)}}} + (1 - \theta) \times \sum_{j=1}^{m} p_{j\cdot l_{(j)}} \quad (13)
\]

Calculate the minimum $F$ of each product in $U$ and take it as the $j$th processed product.

(c) Extract the first two products $\pi(1)$, $\pi(2)$ in the well-sorted product sequence $\pi$, and calculate the $f_1$ values of the two. Then, select the smaller values and reorder them, which is denoted as $\pi^*$. 175
(d) Perform an MME operation on the sorted sequence \( \pi^* \) to generate an initialized population.

The NEH step was basically the same as the MME, and it was used to initialize the population of \( f_2 \).

3.2 Individual generation method optimization

The individual generation method of the traditional ABC algorithm has the disadvantages of uncertainty and slow convergence. In this paper, the generation method of individual in ABC algorithm was improved by introducing the similar block order crossover (SBOX) algorithm into ABC algorithm and using the SBOX algorithm to change the calculation method of crossover operator.

![Fig. 2 Individual generation process of similar block order crossover algorithm](image)

An example was given to illustrate the process of generating a new solution by the SBOX crossover operator, as shown in Figure 2.

1. Given that there are five non-dominated solutions \( \pi_1-\pi_5 \) in the reserve set, they are expressed as:

\[
\begin{align*}
\pi_1 &= \{2, 6, 3, 1, 4, 5, 7\} \\
\pi_2 &= \{3, 5, 1, 6, 7, 2, 4\} \\
\pi_3 &= \{4, 5, 2, 1, 7, 3, 6\} \\
\pi_4 &= \{4, 1, 7, 2, 6, 3, 5\} \\
\pi_5 &= \{4, 5, 2, 6, 3, 7, 1\}
\end{align*}
\]

(14)

Thus, a temporary solution set TS is generated according to the solution \( \pi_1-\pi_5 \), and by calculating the frequency of the workpiece \( i \) sorted before the workpiece \( j \), the maximum value is selected as the product block;

2. Select two parent individuals, Parentk (k=1, 2), and compare the same product blocks of the parent product and the temporary solution set. The same product blocks in the two parent individuals and temporary solution set are inherited to Offs1 and Offs2;

3. Other products in Offs1 and Offs2 are obtained from the parent individual according to the crossover operation.

Due to its characteristics free from position constraints, the SBOX algorithm can ensure the integrity of the product to the greatest extent when generating individuals within the population.

![Fig. 3 The generation process of neighbourhood solution based on PLS algorithm](image)
After generating the new excellent global solutions at the hired bee stage of the ABC algorithm through the above process, the Pareto local search method was used to generate more neighbourhood solutions. The specific process is shown in Figure 3.

Thus, a temporary solution set TS is generated according to the solution $\pi_j$, and by calculating the frequency of the workpiece $i$ sorted before the workpiece $j$, the maximum value is selected as the product block;

(2) Select two parent individuals, Parentk $(k=1, 2)$, and compare the same product blocks of the parent product and the temporary solution set. The same product blocks in the two parent individuals and temporary solution set are inherited to Offs1 and Offs2;

(3) Other products in Offs1 and Offs2 are obtained from the parent individual according to the crossover operation.

Due to its characteristics free from position constraints, the SBOX algorithm can ensure the integrity of the product to the greatest extent when generating individuals within the population.

After generating the new excellent global solutions at the hired bee stage of the ABC algorithm through the above process, the Pareto local search method was used to generate more neighbourhood solutions. The specific process is shown in Figure 3.

Figure 4 shows the implementation of the neighbourhood solution based on the PLS algorithm.

### 3.3 Improved ABC algorithm

Based on the above proposed population initialization optimization method and individual generation strategy within the population, the traditional ABC algorithm was improved. The specific flow of the improved algorithm is shown in Fig. 5.

![Fig. 4 Execution process of neighbourhood solution based on PLS algorithm](image)

![Fig. 5 Improved ABC algorithm flow](image)
(a) Set the population size, crossover operator probability, PLS probability, and iteration termination time;
(b) Use the heuristic algorithms MME and NEH to initialize the population and calculate the objective function value;
(c) After generating a new excellent global solution based on SBOX, use PLS algorithm to generate more neighbourhood solutions;
(d) Reconstruct the new solution and perturb the un-updated solution;
(e) Extract the optimal non-dominated solution obtained from each iteration, and construct a reserve set EAs until the non-dominated solution set meets the calculation termination condition or reaches the maximum number of iterations.

It can be seen from the algorithm flow of Figure 5 that the improved ABC algorithm has significant improvements in population initialization, global solution and neighbourhood solution generation.

4 SIMULATION TEST AND RESULT ANALYSIS

The improved algorithm (IABC) proposed in this paper was validated by comparing it with other improved algorithms such as the NGA algorithm, improved NSGA algorithm (INSGA-II), the block-based estimation of distribution algorithm (BBEDA), and the threshold acceptance algorithm (TA) (Chang, 2015; Han et al., 2014; Marimuthu et al., 2009; Ventura and Yoon, 2013).

It’s assumed that the product quantity set is \( n=\{10, 30, 50, 70, 90, 110\} \); the number of machines is \( m=\{5, 10, 15, 20\} \); the initial population size is 20, the crossover probability is 0.6. PLS=0.4, EAs=100, and the maximum number of iterations is 30.

Firstly, the influence of PLS on the ABC algorithm is analysed. Figure 6 shows the Pareto frontier of the IABC algorithm and the ABC algorithm without PLS (ABC-nPLS) in the 4 randomly selected examples.

It can be seen from Fig. 6 that the convergence of the Pareto frontier calculated by the IABC is significantly superior to that of the ABC-nPLS, indicating that the local search ability of PLS can significantly increase the computational convergence speed of ABC algorithm and help the initial population to evolve more efficiently to the real Pareto frontier.

![Fig. 6 Effect of PLS on the Pareto frontier obtained by the ABC algorithm](image-url)
The above five algorithms were evaluated using the non-dominated solution number N(S) and the non-dominated solution ratio R(S); N(S) can be expressed as

\[ N(S) = \left\{ x' \in S \mid \exists x < \forall y, y \in S \land x \neq y \right\} \]

\[ S = \bigcup_{i=1,2,3,4} S_i \]

The smaller the N(S) is, the less the number of non-dominated solutions calculated and the worse the performance of the algorithm.

R(S) can be expressed as:

\[ R(S) = \frac{N(S)}{|S|} \]  

The larger R value indicates the quality of the non-dominated solution calculated by the algorithm is better.

D(S) represents the distance between S and the reference solution, and it's expressed as:

\[ D(S) = \frac{1}{|S|} \sum_{x \in S} d_x(S) \]  

\[ d_x(S) = \min_{y \in S} \left\{ \sum_{i=1}^{4} \left( f_{x}^{i}(x) - f_{y}^{i}(y) \right)^2 \right\} \]

The larger the D(S) value, the farther the obtained solution set is from the optimal Pareto front.

Table 1 lists the calculation results of the N(S) and R(S) value for the five algorithms mentioned above. It can be seen from the table that under the same problem scale and calculation time, the number of non-dominated solutions was 9 for IABC, 0 for NGA, 2 for INSGA-II, 3 for BBEDA, and 1 for TA. The number of non-dominated solutions in IABC algorithm is much larger than other algorithms.

On the non-dominated solution ratio R(S), the average value was 93% for IABC algorithm, 0% for NGA, 44% for INSGA-II, 38% for BBEDA, and 14% for TA non-dominated solution. That is, more than 93% solutions the algorithm proposed in this paper are not dominated by the non-dominant solutions of other computational methods. Therefore, the quality of the non-dominated solution obtained by the algorithm under the same problem scale and calculation time is higher.

<table>
<thead>
<tr>
<th>Problems</th>
<th>N(S)</th>
<th>R(S) (%)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50×5</td>
<td>0.38</td>
<td>48.00</td>
<td>98.00</td>
</tr>
<tr>
<td>50×10</td>
<td>0.38</td>
<td>48.00</td>
<td>98.00</td>
</tr>
<tr>
<td>50×20</td>
<td>0.45</td>
<td>70.00</td>
<td>98.00</td>
</tr>
<tr>
<td>70×5</td>
<td>1.12</td>
<td>27.00</td>
<td>100.00</td>
</tr>
<tr>
<td>70×10</td>
<td>1.11</td>
<td>22.00</td>
<td>90.00</td>
</tr>
<tr>
<td>70×20</td>
<td>0.23</td>
<td>22.00</td>
<td>90.00</td>
</tr>
<tr>
<td>90×5</td>
<td>2.15</td>
<td>100.00</td>
<td>98.00</td>
</tr>
<tr>
<td>90×10</td>
<td>0.19</td>
<td>75.00</td>
<td>100.00</td>
</tr>
<tr>
<td>90×20</td>
<td>0.16</td>
<td>79.00</td>
<td>98.00</td>
</tr>
<tr>
<td>110×5</td>
<td>1.18</td>
<td>27.00</td>
<td>100.00</td>
</tr>
<tr>
<td>110×10</td>
<td>0.09</td>
<td>50.00</td>
<td>98.00</td>
</tr>
<tr>
<td>110×20</td>
<td>0.37</td>
<td>100.00</td>
<td>89.00</td>
</tr>
<tr>
<td>Mean</td>
<td>0.74</td>
<td>45.38</td>
<td>38.75</td>
</tr>
</tbody>
</table>

Table 1. Calculation results of N(S) and R(S) values for five algorithms

![Graph (a) 70×5](image)

![Graph (b) 70×10](image)
Figure 7 shows the evolution curves of five algorithms at different problem scales according to Equations 17 and 18. It can be seen from the figure that compared with the other four algorithms, the evolution curve of the IABC algorithm is the smallest, which indicates that with the increase of the number of iterations, the optimal solution set can be evolved to the correct Pareto frontier more quickly using the IABC algorithm.

Figure 8 shows the Pareto frontier curves using this algorithm at different problem scales. It can be seen from the figure that as the number of iterations increases, the utilization degree of the non-dominated solutions and the evolution rate of the population, and the scale of the high-quality solutions also increase gradually.

Thus, the proposed algorithm was validated, and the global search and local development performance of the traditional ABC algorithm were also improved.
5 CONCLUSIONS

In this paper, a multi-objective flow shop optimization scheduling model of production workshop considering blocking effect was established. Then, the ABC algorithm was improved to be used in the establishment of the model calculation. Finally, the optimized scheduling method proposed in this paper was verified by simulation examples. The research conclusions are as follows:

(1) The heuristic algorithm MME and NEH were used to generate the initial population. After generating the new excellent global solution based on the SBOX algorithm in the hired bee phase, the PLS algorithm was adopted to generate more neighbourhood solutions, and significantly increase the computational convergence rate of ABC algorithm, which can help the initial population to evolve more efficiently toward the real Pareto frontier.

(2) Simulation test results show that the IABC algorithm proposed in this paper can obtain high-quality non-dominated solutions under the same problem scale and calculation time, and with the number of iterations increasing, the utilization degree of non-dominated solutions and the evolution rate of populations, and the scale of high-quality solutions gradually increase.

6 ACKNOWLEDGEMENTS

Educational Science Planning in Hubei Province, Project No 2018GB064, Research on STEAM Education in Information Technology Course of Primary and Secondary Schools under the Background of Artificial Intelligence.

7 REFERENCES


