FLEXIBLE JOB SHOP SCHEDULING OPTIMIZATION BASED ON NEH HEURISTICS

Dapeng SANG
Department of Business Management, School of Management, Xiamen University, Xiamen 361005, China
Email: azhsjc@163.com

ABSTRACT: Reasonable job shop scheduling is the foundation and key for improving comprehensive strength of the manufacturing company. This paper aims to develop a flexible job shop scheduling optimization method. A mathematical model is built based on literature analysis, simulation experiment and comparative analysis. After improving the standard Bacterial Foraging Optimization (BFO), an improved BFO based on NEO heuristics is proposed. The three algorithms can solve typical problems occurred when operating the flexible job shop scheduling process. The findings show that the convergence speed and stability of the algorithm proposed herein are all superior to that of the other two. Better still, it can also obtain an optimal solution to the problem. This study provides the clues to solving the flexible job shop scheduling problem with a certain value in theory.

KEY WORDS: NEH heuristic method, bacterial foraging algorithm, flexible job shop scheduling

1 INTRODUCTION

The manufacturing industry, as the pillar of the national economy, can directly respond to the productivity level of a country. Faced with diversified demands and fierce competition in the global market, many companies rise to the challenges about how to build the advanced production and manufacturing models, reduce the production costs, and increase the production efficiency (Liu et al., 2017). Shop floor scheduling is the key and base module for improving the comprehensive strength of the company. Reasonable scheduling plan can achieve a rational allocation for resources and help reduce the production cycle of the workpieces (Wang et al., 1996). In response to personalized and diversified requirements from customers, the flexible job shop scheduling problem has become the prevalent one that most people concern in the manufacturing industry.

At the earliest time, the study of the flexible job shop scheduling problem mainly focused on the assembly line balancing. It was upon the "machine scheduling problem" that scholars relied at that time (Moccellin, 1995). Salvadorzai first proposed the flexible flow shop scheduling problem in 1973 (Laha and Sapkal, 2014). Subsequently, many scholars have attempted to tackle the flexible flow shop scheduling problem with the total three types of algorithms, i.e. intelligent optimization algorithm, accurate algorithm and heuristic method (Braglia and Grassi, 2009). Small-scale flexible flow shop scheduling introduces the accurate algorithm to obtain an optimal but poor-quality solution. The intelligent optimization algorithm can work out an optimal solution for large-scale scheduling problems in the flexible flow shops within a shorter time based on the computer and intelligent computing technologies (Taillard, 1990). There are still many relevant methods, which all fail to effectively solve such problem. In 2002, Passino proposed a bionic random algorithm, i.e. the Bacterial Foraging Optimization (BFO) (Sheibani, 2010). Thanks to its extremely strong optimization capacity, it has aroused common concern of scholars. Now the study of BFO at home and abroad focuses on three dimensions, i.e. the algorithm theory, improvement and practices (Liang et al., 2011). However, as the proposal of the algorithm is relatively later, the study still lingers in its infancy so that it has not yet borne some fruits.

Based on the above analysis, when roughly analyzing relevant theory of flexible job shop scheduling problem, a mathematical model should be built for flexible job shop scheduling problem. After the standard BFO has been improved, an IBFO-NEH is then proposed against the typical flexible flow shop scheduling problem to be solved. Three types of algorithms are used to solve the Car-like problem for typical flexible flow shop scheduling. The comparative analysis of the results bears out that the IBFO-NEH enables to improve the flora quality, accelerate the algorithm convergence, and obtain the optimal solution of the problem.
2 FLEXIBLE JOB SHOP SCHEDULING PROBLEM

2.1 Mathematical model

The flexible job shop scheduling problem can be interpreted as follows (Ben-Daya and Al-Fawzan, 1998): In order to optimize the processing performance indicator \( C_{\text{max}} \), the machining sequence is determined for \( n \) workpieces with the identical process route on \( m \) machines. Each workpiece can only be machined on one machine, and one machine can only process one workpiece at a time. Assume all workpieces are machined parallelly at zero time, regardless of the machining priority of the workpieces, the machine can work continuously, for easy computation, simplify the flexible job shop scheduling problem and translate it into the following mathematical model (Jolai et al., 2012):

Decision variables:

\[
X_{i,k} =
\begin{cases}
1, & \text{If the workpiece } j \text{ is ranked at the } kth \text{ position in the sequence } R \\
0, & \text{Other situations}
\end{cases}
\] (1)

Objective function:

\[
\min(C_{\text{max}}) = \min \left( \max_{k \in [1,2,K,n]} \left( C_k, n \right) \right) \] (2)

Constraint condition:

\[
\sum_{i=1}^{n} X_{i,k} = 1, i \in \{1, 2, K, n\} \] (3)

\[
\sum_{k=1}^{n} X_{i,k} = 1, k \in \{1, 2, K, n\} \] (4)

\[
C_{1,1} = \sum_{i=1}^{n} X_{i,1} g_{i-1} \] (5)

\[
C_{k+1,j} \geq C_{k,j} + \sum_{i=1}^{n} X_{i,k+1} g_{i,j} \] (6)

\[
k \in \{1, 2, K, n - 1\}, j \in \{1, 2, K, m\} \]

\[
C_{k,j+1} \geq C_{k,j} + \sum_{i=1}^{n} X_{i,k} g_{i,j+1} \] (7)

\[
k \in \{1, 2, K, n\}, j \in \{1, 2, K, m - 1\} \]

Where: \( C_{k,j} \) is the completion time of the workpiece \( k \) machined on the machine \( M_j \); \( t_{i,j} \) is the machining time of the process.

2.2 Evaluation indicator for flexible job shop scheduling problem

In the production scheduling process, customer satisfaction, economic indicators, and maximum and minimum capacities are often used to measure how well the scheduling program works (Fattahi et al., 2013), where the maximum process time \( C_{\text{max}} \) (Li et al., 2013) means it operates until the time when the last process is actually ended. Calculated by the formula (9), it is one of the commonly used machining performance indicators in the flexible job shop scheduling. Improved maximum process time can help to improve production efficiency. This paper uses a forward calculation method to obtain the maximum process time for each workpiece.

\[
C_{\text{max}} = \max \{ C_{i,1} | 1 \leq i \leq n \} \] (9)

2.3 Flexible flow shop scheduling problem

There are three types of commonly used flexible job shop scheduling methods as follows (Cheng et al., 1996):

**Heuristic method**

The heuristic method features short calculation time, but the solution it obtains is sub-optimal. The maximum machining time \( C_{\text{max}} \) is commonly used as an evaluation indicator. NEH, RA and CDS are several algorithms used to solve flexible job shop scheduling problem.

**Intelligent optimization algorithm**

The intelligent optimization algorithm was developed based on computer technology, while the swarm intelligence optimization algorithm was proposed as an intelligent type on the principle of the bionics, including particle swarm optimization algorithm, ant colony optimization algorithm, artificial fish swarm optimization algorithm, and artificial bee colony optimization, bacterial foraging algorithm (Fattahi et al., 2013). All of these can quickly and reliably solve the optimal solution in the absence of global information.

**Accurate algorithm**

This method applies to small-scale flexible shop scheduling problems. Although the optimal solution can be obtained, it consumes too long calculation time so that it is difficult to adapt to the practical situation of the manufacturing shop floor. The commonly used accurate algorithms include the...
enumeration method, the branch and bound method and so on.

3 FLEXIBLE JOB SHOP SCHEDULING OPTIMIZATION BASED ON NEH HEURISTICS

3.1 Standard BFO

The standard BFO is a swarm intelligence optimization algorithm designed after simulating the foraging behaviors of E. coli in the human intestine, including three operation cycles of tropism, cloning, and migration (Li et al., 2013). Flow chart of the standard BFO is shown in Figure 1.

3.2 IBFO

Since the standard BFO does not allow for the interaction between bacteria during the operation, the individuals with good fitness easily escape if the health assessment criteria are used. The Original individuals are replaced by randomly generated bacteria. All these problems affect the convergence rate and optimization effect of algorithm (Cheng et al., 1996). For this purpose, this paper proposes the following improvements to the standard BFO algorithm:

(1) In order to speed up the convergence speed of the algorithm, after the tropism operation, a crossover optimization operator is executed, and the partial cross-mapping is used to guide the flora to the optimal area with the information of the optimal individuals.

(2) In order to make sure that individuals with the best fitness in the population can be reserved, a hybrid cloning strategy based on fitness and health rankings is preferred.

(3) Unlike the standard BFO which adopts different fixed migration probability, all flora in the IBFO migrate according to the adaptive migration probability. The calculation formula is shown in (10) (Daniels and Mazzola, 1993).

\[
P_{v-ed} = \left(1 - \frac{H^i - H_{min}}{H_{max} - H_{min}} \right) \cdot \frac{J^i - J_{min}}{J_{max} - J_{min}} = P_{ed}
\]

The flow of the IBFO is shown in Figure 2.
3.3 Design of IBFO based on NEH heuristics (IBFO-NEH)

Problem coding, algorithm initialization, fitness function, operation operator, and algorithm termination condition are designed as five procedures of BFO to solve the flexible job shop scheduling problem (González-Neira et al., 2016), as shown in Figure 3.

3.4 Results and Analysis of IBFO for Solving Car-like Problems

In order to test whether the standard bacterial foraging algorithm (BFO), IBFO and NEH-based IBFO are effective against the flexible shop scheduling problems, this paper uses three algorithms to solve typical car-like problem (Ruiz et al., 2008) and compares their results.

It can be seen from the comparative results of the three algorithms shown in Tables 1 and 2 that the three can obtain the optimal solutions to the problem, which shows that they enable to solve the flexible shop floor scheduling problem. It can be seen from the comparison of Num values that the global random search capacities of the three algorithms in descending order are IBFO> IBFO-NEH> BFO, and the relative errors of the three algorithms are IBFO-NEH> IBFO> BFO.
respectively. It shows that the stability of IBFO-NEH is better than the other two algorithms.

Table 1. Solution results of three algorithms

<table>
<thead>
<tr>
<th>Scheduling problem (n*m)</th>
<th>BFO</th>
<th>IBFO</th>
<th>IBFO-NEH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C^*_BFO$</td>
<td>Num</td>
<td>$C^*_IBFO$</td>
</tr>
<tr>
<td>Car1 (11*6)</td>
<td>7036</td>
<td>3</td>
<td>7036</td>
</tr>
<tr>
<td>Car2 (12*4)</td>
<td>7156</td>
<td>4</td>
<td>7156</td>
</tr>
<tr>
<td>Car3 (15*5)</td>
<td>7302</td>
<td>1</td>
<td>7302</td>
</tr>
<tr>
<td>Car4 (11*7)</td>
<td>8013</td>
<td>2</td>
<td>8013</td>
</tr>
<tr>
<td>Car5 (17*9)</td>
<td>7621</td>
<td>1</td>
<td>7621</td>
</tr>
<tr>
<td>Car6 (10*5)</td>
<td>8405</td>
<td>3</td>
<td>8405</td>
</tr>
<tr>
<td>Car7 (8*7)</td>
<td>6588</td>
<td>2</td>
<td>6588</td>
</tr>
<tr>
<td>Car8 (9*9)</td>
<td>8345</td>
<td>1</td>
<td>8345</td>
</tr>
</tbody>
</table>

Where, n*m represents the number of workpieces * the number of machines, Num represents the maximum number of optimal processing programs searched in the operation, which can reflect the global search capacity of the algorithm.

Table 2. Comparison of three algorithm results

<table>
<thead>
<tr>
<th>Scheduling problem (n*m)</th>
<th>BFO</th>
<th>IBFO</th>
<th>IBFO-NEH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARE</td>
<td>BRE</td>
<td>WRE</td>
</tr>
<tr>
<td>Car1 (11*6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car2 (12*4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car3 (15*5)</td>
<td>0.59%</td>
<td>0</td>
<td>1.07%</td>
</tr>
<tr>
<td>Car4 (11*7)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car5 (17*9)</td>
<td>0.26%</td>
<td>0</td>
<td>0.55%</td>
</tr>
<tr>
<td>Car6 (10*5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car7 (8*7)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car8 (9*9)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Where, ARE, BRE, WRE represent the average, minimum and maximum relative errors, respectively.

Figure 4-6 shows an optimization solution for the three algorithms to the Card6 problem. It is found from the comparison that the two IBFOs have a number of iterations of 12, while the standard BFO is 6, which is because the IBFOs introduce a cross-optimal operator after the tropism operation. As shown in Figure 2 and 3, two IBFOs converge faster than the standard BFO in Figure 1. Compare the initial solution qualities of the thalli in Figure 2 and Figure 3, it can be seen that the IBFO based on NEH heuristics is higher than the common one. The initial solutions of the two algorithms are 8763 and 9160, respectively, and the convergences are improved.

Hit-goal represents the probability that the algorithm will reach the optimal solution, which can be calculated by the formula (11) (Attar et al., 2013):

$$P_{hit-goal} = \frac{\text{Number of times the algorithm gets the optimal solution}}{\text{Number of times the algorithm is run}}$$

Figure 4. A search optimization solution for solving the Card6 problem by BFO algorithm
4 CONCLUSION

The flexible job shop scheduling optimization is one of problems most widely investigated by manufacturing companies. This paper attempts to apply the BFA to the flexible job shop scheduling problems. Several conclusions come here as follows:

(1) Based on the study of standard BFO, it should be improved from the tropism, cloning and migration operations, and the flow of the improved BFO should also be designed in detail.

(2) In order to further optimize the flexible job shop scheduling problem, the NEH heuristics and the improved bacterial foraging optimization (IBFO) are integrated to propose an IBFO-NEH.

(3) Car-like problems in flexible job shop scheduling are simulated by an experiment using BFO, IBFO and IBFO-NEH. The laboratory findings show that the IBFO-NEH algorithm can effectively solve the flexible flow shop scheduling problem and obtain the optimal solution. It indeed has a certain practical application value in the manufacturing industry.

5 REFERENCES


