LOW-CARBON FLEXIBLE JOB-SHOP SCHEDULING BASED ON IMPROVED NONDOMINATED SORTING GENETIC ALGORITHM-II


* School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
** Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou 310018, China
E-Mail: sengdw@163.com (Corresponding author)

Abstract
Considering the impacts of multiple production objectives, makespan and low carbon factor on job-shop scheduling optimization, this paper puts forward a novel low carbon scheduling method for flexible job-shop based on the improved nondominated sorting genetic algorithm-II (NSGA-II). Firstly, a low-carbon scheduling optimization model was established for multi-objective, multi-speed job-shop. Then, the flow of the NSGA-II-based core algorithm was explained, and the new population selection was optimized through the calculation of congestion and nondominated level. Finally, multiple simulation examples were adopted to validate the proposed algorithm. The results show that the proposed NSGA-II low carbon optimization algorithm can converge to the global best Pareto solution rapidly, and lower the no-load and total energy consumption of the production line through automatic management while ensuring production efficiency.

Key Words: Flexible Job-Shop Scheduling Problem (FJSP), Nondominated Sorting Genetic Algorithm-II (NSGA-II), Low-Carbon Scheduling

1. INTRODUCTION

Currently, almost all industries are competing to achieve sustainable development through energy conservation and emission reduction. To protect the environment and improve enterprise profitability, it is necessary to reduce the carbon emissions and consume less energy in the production process [1, 2].

From automobile industry to chemical production, most processing and manufacturing industries rely on the intensive flow operation mode. In this mode, each job-shop, involving multiple processes with similar techniques, can be viewed as a flexible job-shop [3-7]. The scheduling of the job-shop is essential to the quality of the products.

Much research has been done on flexible job-shop problems (FJSP) from the perspective of operational scheduling. The objectives of these studies fall into three categories: (1) optimizing the energy consumption through low-carbon scheduling [8-10]; (2) saving energy and reducing emissions through on/off scheduling, i.e. timely shutting down idle machines with long waiting time, in addition to the conventional configuration of the start/completion time for the machine in each process [11, 12]; (3) saving energy and reducing emissions by reducing the machine speed in time without affecting the production scheduling. With energy saving and emission reduction as the scheduling target, the existing studies on the FJSP cover single-machine scheduling, parallel-machine scheduling, flow shop scheduling and job-shop scheduling.

Single-machine scheduling: Che et al. explored multi-speed single-machine scheduling problem with limited delivery time, established a mixed integer linear programming model for minimum energy consumption, and solved the model with the CPLEX software [13]. Angel et al. probed into the multi-speed dynamic single-machine scheduling problem with...
limited energy, set up an optimization model for maximum throughput, and proposed a
dynamic programming algorithm [14].

Parallel-machine scheduling: Cataldo et al. studied the energy efficiency in parallel
machine scheduling, created a hybrid logic dynamic scheduling model, and solved the model
with Matlab toolbox [15]. Focusing on parallel-machine scheduling with a limit on the
maximum power, Wang et al. built an integer programming model targeted at the makespan,
solved the model by the genetic algorithm, and suggested that the makespan should be
minimized using a fast cutting speed without exceeding the power limit [16]. Albers and
Schmelzer investigated the dynamic parallel-machine scheduling problem with deadlines,
established a variable speed model for minimum energy consumption, and solved the model
with polynomial time algorithm [17].

Flow shop scheduling: Fang et al. studied the multi-speed flow shop problem, built an
integer programming model with makespan, peak power and carbon emissions as the
objectives, and solved the model with commercial planning software [18]. These scholars put
forward the concept of continuous speed after discussing the discrete speed problem.
Mansouri et al. examined the flow shop scheduling problem with two machines of different
speeds, established a multi-objective mixed integer linear programming model, and developed
a heuristic method to solve the problem [19]. Using the genetic simulated annealing
algorithm, Dai et al. solved the flexible flow shop problem targeted at the makespan and
energy consumption, and determined if the machine is shutdown or unloaded in the idle state
[20]. Considering the on/off of the machine, An and Yan tackled the machine energy
consumption of flexible flow shop by selecting the spindle speed of the machine, developed a
mixed integer programming model focusing on makespan and energy consumption, and
adopted genetic-simulated annealing algorithm to solve the problem [21]. Tang et al. looked
into the dynamic flow shop scheduling problem, found an energy-saving model focusing on
makespan and energy consumption, and solved the problem with an improved particle swarm
optimization algorithm [22]. Moreover, Li et al. dealt with the energy-saving flexible process
planning problem and the integration of process planning and shop scheduling, and pursued
the goals of makespan and energy consumption using the improved genetic algorithm [23].

Job shop scheduling: Zhang and Chiong attempted to save energy of the job shop by
changing the processing speed, established a mathematical model focusing on the total
weighted delay and total energy consumption, and proposed a hybrid genetic algorithm with
local search strategy to solve the problem [24]. Salido et al. studied the JSP with speed
scaling, created a mathematical model with makespan and energy consumption as the
objectives, and presented a genetic algorithm to solve the problem [25]. Furthermore, Zhang
et al. dug into static scheduling and dynamic scheduling problems in flexible manufacturing
systems, created a mathematical model for minimum makespan and minimum or maximum
energy consumption, and solved the problem with linear programming software [26].

To sum up, researchers have just begun to consider energy saving and emissions reduction
of traditional scheduling models, and most of them emphasized on single-machine, parallel-
machine and flow shop scheduling problems rather than the FJSP. Facing the problems in the
existing studies, this paper probes into the low-carbon scheduling of multi-objective flexible
job-shop based on the improved nondominated sorting genetic algorithm-II (NSGA-II), and
comprehensively considers the effects of multiple production objectives, make-span and low-
carbon factor on job-shop scheduling optimization [27]. The research findings lay a
theoretical basis for the scheduling of new job-shops.
2. MULTI-OBJECTIVE, MULTI-SPEED JOB-SHOP SCHEDULING OPTIMIZATION MODEL

The research object is a multi-objective, multi-speed job-shop containing \( m \) machines and \( n \) jobs. Each machine can control its processing speed. The process of the \( i^{th} \) job is denoted as \( n_i \). The following hypotheses were put forward to simplify the calculation.

(a) All machines, jobs and materials in the job-shop are ready at \( t = 0 \);
(b) The actual processing sequence of the jobs is consistent with the planned sequence;
(c) Each machine can process a job at any processing speed, but the energy consumption differs with the processing speed;
(d) Each machine can only process one job at a time, and each job can only be processed on one machine.

The temporal variation of the processing power of a machine is presented in Fig. 1, where \( t_1 \) and \( t_7 \) are the shutdown periods of the machine and \( Q_{gk} \) is the energy consumption of the machine in the on/off states.

\[
Q_{gk} = \int_a^{a+T_{i,j}} P_k(t)\, dt + \int_{b-T_{i,j}}^b P_k(t)\, dt \max_{i,j,q} \left( X_{ijk} \right);
\]
\[
a = \max_{i,j,q} \left( X_{ijk} \left( C_{ijk} - T_{ijk} \right) \right), \quad b = \max_{i,j,q} \left( X_{ijk} C_{ijk} \right)
\]

Besides, \( t_3 \) and \( t_5 \) are the no-load periods of the machine. In these periods, the energy consumption of the machine can be expressed as:

\[
Q_{dk} = \sum_q Z_{kq} \left\{ \sum_i \max_{n_i} \left[ Y_{ijk} \left( C_{ijk} - C_{hik} - T_{ijk} \right) X_{ijk} X_{hik} \right] \right\}
\]

Moreover, \( t_2, t_4 \) and \( t_6 \) are the processing periods of the machine. In these periods, the energy consumption of the machine can be expressed as:

\[
Q_{ck} = \beta \sum_q P_{kq} \sum_i \sum_j X_{ijk} T_{ijk}
\]

It can also be seen that the machine is in standby state between 0 and \( t_1 \) and after \( t_7 \). In this case, the energy consumption of the machine can be expressed as:

\[
Q_{fk} = C_{\max} Z_{fk}
\]

According to Eqs. (1) to (4), the total energy consumption of the machine throughout the processing process can be calculated as:

\[
Q_k = Q_{gk} + Q_{dk} + Q_{ck} + Q_{fk}
\]
The established model must achieve the minimum makespan $C_{\text{max}}$ and minimum energy consumption $Q_k$, that is:

$$f = \min\{C_{\text{max}}, Q\}$$  \hfill (6)

$$C_{\text{max}} = \max_{i,j} X_{ijkq} C_{ijkq}$$  \hfill (7)

$$C_q - T_{ij} \geq C_{ij}(j-i)$$  \hfill (8)

$$\left(\frac{Y_{ijkq}}{2}\right)\left(Y_{ijkq} - 1\right)(C_{kj} - C_q - T_{jklq}) X_{ijklq} X_{jklq} +$$

$$\left(\frac{Y_{ijkq}}{2}\right)\left(Y_{ijkq} + 1\right)(C_q - C_{kl} - T_{ijklq}) X_{ijklq} X_{klijq} \geq 0$$  \hfill (9)

$$\sum_{k=1}^{m} X_{ijkq} = 1$$  \hfill (10)

Where $X_{ijkq}$ and $C_{ijkq}$ are the decision variables of $C_{\text{max}}$. Eqs. (8) to (10) are the constraints on $C_{\text{max}}$ and $Q_k$.

### 3. FJSP OPTIMIZATION ALGORITHM BASED ON IMPROVED NSGA-II

The FJSP optimization algorithm was established based on the improved NSGA-II. The flow of the algorithm is shown in Fig. 2 below.

Specifically, the scheduling is optimized in the following process:

1. Set such parameters as the number of iterations and the population size, and initialize the model;
2. Calculate fitness;
3. Select excellent child individuals based on crossover and mutation probabilities, and form a new population;
4. Calculate the minimum makespan and total carbon emissions of the new population, and perform nondominated sorting of the population;
5. Determine the number of new individuals with a nondominated level of 1. If the number is greater than $P_{\text{size}}$, calculate the congestion of all new individuals, and relocate the individuals with a small congestion into the next population; otherwise, calculate the congestion of individuals with a nondominated level of 2;
6. Determine the optimal individual in the current population.

Before each processing, the machine and power were assigned adaptively to each job by the “gap extrusion method” in Fig. 3. The basic principle is to minimize the energy consumption.

The core of the gap extrusion method lies in the full utilization of the gap between the arranged processes. If the immediate predecessor process has a shorter makespan than the previous process on a machine available to a process, it is necessary to consider whether the machine has any gap in this period. The gap should be prioritized if the current process can be inserted into it. For instance, process $O_{3,1}$ need to be arranged after the arrangement of processes $O_{5,1}$, $O_{5,2}$, $O_{4,1}$, $O_{4,2}$ and $O_{5,3}$. If process $O_{3,1}$ is processed on machine 3, check if there is a gap before this process; if yes, check whether the gap size allows the insertion of the process; if the process can be inserted, process $O_{3,1}$ should be arranged before process $O_{5,3}$.

In actual allocation, the machine running at a low processing speed should be preferred. Before selection, the energy consumption, processing time and machine load should be calculated for each process in advance.
4. EXPERIMENTAL ANALYSIS

To prove the feasibility and superiority of our model, the ten standard examples, MK01~MK10, constructed by Brandimarte were adopted for the validation. Fig. 4 presents the energy consumption of the Pareto solutions to MK01 at different makespans. It is clear that the relationship between makespan and energy consumption in the Pareto solutions obeyed the exponential distribution, indicating that the two factors are well balanced.
Figure 4: Energy consumption of the Pareto solutions at different makespans.

The Gantt chart of the Pareto solutions to MK01 is given in Fig.5, where the x-axis is the processing time and the y-axis is the machine. Six machines are illustrated in this figure. The blocks in different colours represent the different processing sequences. The darker the colour, the greater the processing power of the machine for the job. The number above the block chain is the serial number of jobs, while that below the chain is the serial number of process. As shown in the figure, machine 2 was the bottleneck of the whole production line, requiring the highest power for processing; machines 1 and 3 were low-load machines, which should use a low power to further reduce the energy consumption of the production line.

Figure 5: Gantt chart of the Pareto solutions to MK01.

Table I lists the Pareto solutions to MK1 with time, energy consumption and low carbon emissions as the objectives. The results show that the time-based decoding algorithm produced the most efficient production plan (the first solution), while the energy consumption decoding algorithm led to the most energy-efficient plan (the third solution); the proposed low carbon scheduling algorithm boasted the advantages of the other two algorithms.
Table I: Pareto solutions to MK01 with time, energy consumption and low carbon emissions as the objectives.

<table>
<thead>
<tr>
<th>Decoding algorithm</th>
<th>Number of Pareto solutions</th>
<th>Pareto solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>6</td>
<td>(48,6.61), (45,6.88), (47,6.74), (46,6.76), (43,6.97), (41,6.70)</td>
</tr>
<tr>
<td>Energy consumption</td>
<td>5</td>
<td>(94,5.96), (88,5.88), (91,5.92), (71,5.87), (78,5.99)</td>
</tr>
<tr>
<td>Low carbon emissions</td>
<td>6</td>
<td>(44,6.91), (45,6.74), (44,6.73), (48,6.43), (48,6.55), (56,6.32)</td>
</tr>
</tbody>
</table>

Fig. 6 compares the energy consumption of the three decoding algorithms, and Fig. 7 displays their energy consumption in processing, no-load and on/off statuses. It can be seen that the proposed scheduling optimization algorithm effectively reduced the total energy consumption and makespan. According to the optimal plan, the makespan was 51 min and the energy consumption was 6.2 kWh. This means the proposed algorithm can decrease the total energy consumption of the production line through automatic management while ensuring production efficiency.

Figure 6: Energy consumption of the three decoding algorithms.

Figure 7: Processing, no-load and on/off energy consumption of the three decoding algorithms.
Next, the three algorithms were further compared in terms of no-load energy consumption and on/off energy consumption. The results of the six machines are listed in Table II. For machine 1, the three algorithms were ranked as: time-based decoding algorithm, low carbon scheduling algorithm and energy consumption decoding algorithm in descending order of no-load energy consumption. For machine 2, the machine, as the bottleneck of the production line, was in full-load operation using any of the three algorithms. For machine 3, the machine was in full-load operation by low carbon scheduling algorithm and time-based decoding algorithm, and mostly (60%) in no-load status by energy consumption decoding algorithm; for machines 4 and 5, machine 5 was in full-load operation by low carbon scheduling algorithm while machine 4 was in full-load operation by time-based decoding algorithm, indicating that the energy consumption differs with the machines for processing. Overall, the proposed low carbon scheduling algorithm can achieve the least no-load energy consumption.

Table II: No-load and on/off energy consumption of machines with time, energy consumption and low carbon emissions as the objectives.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Low carbon scheduling algorithm</th>
<th>Time-based decoding algorithm</th>
<th>Energy consumption decoding algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-load</td>
<td>On/off</td>
<td>No-load</td>
<td>On/off</td>
</tr>
<tr>
<td>Machine 1</td>
<td>56 %</td>
<td>44 %</td>
<td>64 %</td>
</tr>
<tr>
<td>Machine 2</td>
<td>0 %</td>
<td>100 %</td>
<td>0 %</td>
</tr>
<tr>
<td>Machine 3</td>
<td>0 %</td>
<td>100 %</td>
<td>0 %</td>
</tr>
<tr>
<td>Machine 4</td>
<td>30 %</td>
<td>70 %</td>
<td>0 %</td>
</tr>
<tr>
<td>Machine 5</td>
<td>0 %</td>
<td>100 %</td>
<td>42 %</td>
</tr>
<tr>
<td>Machine 6</td>
<td>29 %</td>
<td>71 %</td>
<td>55 %</td>
</tr>
</tbody>
</table>

The proposed algorithm was further compared with the multi-objective simulated annealing algorithm (MOSAA). The initial and final temperatures \(W_0\) and \(W_z\) of the MOSAA can be calculated as:

\[
W_0 = \max \left( \left( \frac{T_{\text{min}} - T_{\text{max}}}{\ln p_r} \right) \left( Q_{\text{min}} - Q_{\text{max}} \right) / \left( \ln p_f \right) \right)
\]

\[
W_z = \min \left( \left( \frac{T_{\text{min}} - T_{\text{max}}}{\ln p_r} \right) \left( Q_{\text{min}} - Q_{\text{max}} \right) / \left( \ln p_f \right) \right)
\]

The temperature update function is:

\[
W_s = \alpha \cdot W_{s-1}
\]

Fig. 8 illustrates the Pareto solutions to MK01~10 obtained by the proposed algorithm and the MOSAA. It can be seen that the proposed algorithm yielded better Pareto solutions than the MOSAA in most of the ten examples. Fig. 9 shows the hypervolumes of the proposed algorithm and the MOSAA for MK01~10 [28]. Table III provides the minimized makespan and minimum energy consumption of the two algorithms. The hypervolume results demonstrate that the proposed algorithm outputted better Pareto solutions than the MOSAA and outperformed the latter in both minimum makespan and minimum energy consumption.
Figure 8: Pareto solutions to MK01–10 obtained by the proposed algorithm and the MOSAA.
5. CONCLUSIONS

Based on the improved NSGA-II algorithm, this paper studies the low-carbon scheduling of multi-objective FJSP, and comprehensively considers the impacts of multiple production objectives, makespan and low carbon factor on scheduling optimization. Firstly, a low-carbon scheduling optimization model was established for multi-objective, multi-speed job-shop. Then, the flow of the NSGA-II-based core algorithm was explained, and the new population selection was optimized through the calculation of congestion and nondominated level. Finally, multiple simulation examples were adopted to validate the proposed algorithm. The results show that the proposed NSGA-II low carbon optimization algorithm can converge to the global best Pareto solution rapidly, and lower the no-load and total energy consumption of the production line through automatic management while ensuring production efficiency.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (No. 61473108), the Public Projects of Zhejiang Province (No. 2013C33082), Zhejiang Provincial Natural Science Foundation (No. LY15F020038 and No. LQ13F020005) and the research foundation of the Education Department of Zhejiang Province (No. Y201430884).
REFERENCES


722


