Apply Hybrid-GSO to Solve Whole-set Orders Scheduling Problem with Setup Time

Xiaomei Yang¹, Jianchao Zen¹,²* and Yu Zhang¹

¹Division of Industrial and System Engineering, Taiyuan University of Science & Technology, Taiyuan 030024, P.R. China
²School of Computer Science and Control Engineering, North University of China, Taiyuan 030051, P.R. China

Abstract

To whole-set orders scheduling problem, different categories of jobs are produced and the set-up time of machines should be considered during the production process. Through combining the theory of whole-set orders scheduling problem with glowworm swarm optimization, a hybrid-GSO is proposed. Improved population strategy and crossover operation is introduced in this algorithm taking into account the characteristics of whole-set orders scheduling problem with setup time. The simulation results validated its feasibility and efficiency.

Key Words: Whole-set orders scheduling, Setup time, Hybrid-GSO

1. Introduction

For many corporations of ‘Make to orders’, a customer’s orders may include multiple workpieces. Since these workpieces have their own deadline, one delayed piece will account for the delivery delay of the whole order. It is called as whole-set order scheduling problem. This problem’s research scope includes job-arrangement and production scheduling, which can better reflect the corporations’ service level and meet customers requirements. Therefore, whole-set order scheduling problem has become an important branch in the field of production schedule and has practical significance.

At present, the scheduling algorithm can be divided into accurate algorithm and approximation algorithm. Accurate algorithm (mathematical programming, branch and bound algorithm, Lagrangian relaxation, etc.) can get accurate solutions of problems. However, it needs large amount of calculation and it is time-consuming which limits its application in solving small-scale problems. Approximation algorithm (genetic algorithm, particle swarm optimization, ant colony algorithm, etc.) have been widely applied in solving large-scale problems because of their simple operation and parallel processing. Glowworm swarm optimization (GSO) is proposed as one of nature-inspired algorithm by Krishnand [1]. It mainly uses the individual glowworm’s characteristics to search partners with its range and move towards better individuals. In this algorithm, the population will focus on one point or some points to undergo the evolution of position. Because GSO has less adjustable parameters and fast convergence rate, it has been widely used in pattern recognition, routing, combinatorial optimization and so on [2–6].

To solve the scheduling problem with orders, Si applied the dynamic programming method to solve the problem of minimizing the number of late orders with multiple job classes [7]. Zhong proposed an order acceptance and scheduling model with machine availability constraints [8]. Wei designed a feature-based order acceptance and scheduling model [9].

Although the above studies have enriched the research achievements of whole-set orders, they all assume that the machines in the process are running continu-
ously which reduced the constraints and complexity of
the production scheduling problem. In the actual produc-
tion, different categories of jobs are produced and the
set-up time of machines can’t be ignored. Compared
with the batch production mode, the whole-set orders
scheduling problem has three characteristics: lower in
workpieces’ quantity, lots of groups and high in the pro-
portion of set-up time. Therefore, it is necessary to re-
search the whole-set orders scheduling problem with
set-up time. Combining with the characteristics of this
scheduling problem and glowworm swarm optimization
(GSO), a Hybrid-GSO with improved population strat-
egy and timely crossover operation is proposed in this
paper.

2. Whole-set Orders Scheduling Problem with
Setup Time

Whole-set orders scheduling problem with set-up
time is described as follows: There are totally \( n \) work-
pieces from \( H \) orders. These orders need to be processed
on a machine and they can be divided into \( Z \) groups. Let
\( S_Z \) be the set-up time of \( Z \) th group. Suppose that set-up
time of two adjacent workpieces are needed while their
groups are different. Besides, all the workpieces can be
arbitrary ordered and \( h \)th (\( 1 \leq h \leq H \)) order includes \( q_h \)
jobs. Let \( J_i^h, P_i^h, d_i^h (1 \leq h \leq H, 1 \leq i \leq q_h) \) respectively
indicates workpiece \( i \)th job of \( h \)th order, its processing
time and deadline. \( P_i^h, d_i^h \) are non-negative. At the beginning,
all the workpieces have been ready for work. Let the fin-
ishing time of \( J_i^h \) be \( C_i^h \) and the weight of each order be
\( q(h) \).

Firstly, the whole-set order coefficient \( x_h \) is defined
that
\[
x_h = \begin{cases} 
1, & \sum_{j=1}^{q_h} U_j^h = q_h \\
0, & \text{else}
\end{cases}
\]
Eq. (1) means that if order \( h \)th is on time, \( x_h = 1 \), or else
\( x_h = 0 \).

where
\[
U_j^h = \begin{cases} 
1, & C_j^h \leq d_i^h \\
0, & \text{else}
\end{cases}
\]

The parameters are defined as follows:
\( J_i^h \): workpiece in \( i \)th position
\( h(J_i^h) \): order number of workpiece \( J_i^h \)
\( Z(J_i^h) \): the processing group of \( J_i^h \)
\( S(Z(J_i^h)) \): set-up time of group \( Z \)
\( P(J_i^h) \): processing time of workpiece \( J_i^h \)
\( C(J_i^h) \): finishing time of workpiece \( J_i^h \)
\( d(J_i^h) \): deadline of workpiece \( J_i^h \)
\( DC_h \): finishing time of order \( h \)
\( q(h(J_i^h)) \): weights of each order

To find the maximum solution of whole-set orders
coefficients, the scheduling model is proposed as:

\[
\text{Obj.} \quad O = \max \sum_{h=1}^{H} q(h) \cdot x_h \tag{3}
\]
\[
s.t. \quad C(J_i) = P(J_i) \tag{4}
\]
\[
G(J_i) = \begin{cases} 
S(Z(J_i)), & Z(J_{i-1}) \neq Z(J_i) \\
0, & Z(J_{i-1}) = Z(J_i)
\end{cases} \tag{5}
\]
\[
C(J_i) = C(J_{i-1}) + G(J_i) + P(J_i), i = 2, 3, \ldots, n \tag{6}
\]

In this model, Eq. (3) represents the objective function
of this scheduling problem. Eq. (4) emphasizes the first
workpiece will be processed without set-up time. Eq.
(5) ensures the set-up time of workpieces from different
groups. The finishing time of every workpiece can be
expressed as Eq. (6).

3. Description of Hybrid-GSO

According to the classic GSO, in optimization pro-
cedure of glowworm population, the individual’s move
can be regarded as a relatively independent displacement
transformation. The relationship among each other can
be reflected by fluorescein value. The direction of evolu-
tion is chosen through analysing the characteristics of
their mutual attraction. To the whole-set orders problem,
it is a NP-hard problem. Its objective is to find the opti-
mal processing sequence to maximize the objective func-
tion. Therefore, GSO is suitable to solve whole-set or-
ders scheduling problem with setup time. However, the
classic GSO has some shortcomings. Firstly, the running
time of the algorithm will lengthen as the scale of this
scheduling problem broadened. Secondly, this algorithm is greatly influenced by initial population. Then its local search ability is poorer than other algorithms. In order to find the better solution with GSO, combining with the characteristics of the whole-set orders problem, a Hybrid-GSO algorithm is proposed to solve the whole-set orders problem with setup time. The key points of this algorithm are:

(1) Generating of initial population

To find the optimum scheduling solution, Earliest Due Date (EDD) rule is used in this paper [10]. Through the analysis of the scheduling results, we found that the same order artifacts in the optimal sequence have a high probability to meet EDD rule. So initial population is divided into the same amount of two parts, where the first part is completely randomized and the second part is generated based on EDD rule. Here is an example: there are 5 work pieces, namely 1, 2, 3, 4, 5, and deadlines ascend gradually. 1 and 3 is from order one while 2 and 5 is from order two. 4 is from order three. After decoded, if sequence 5/45/3/45/1 is obtained, the workpieces should be rearranged according to the rule from same order because the sequence is not obeying the ascending deadline rule. So the sequence 2–5–1–3–4 is obtained finally through rearranging. This kind of population with the above rule has a high probability bias towards the optimal individual.

(2) Step strategy

The moving step of this algorithm is designed as the Euclidean distance between two individuals. This method is helpful to avoid infeasible individuals and premature convergence.

(3) Foraging behavior

Foraging behavior comes from artificial fish-swarm algorithm (AFSA) which is proposed by Li [11]. It has been verified that it can well solve the problem of nonlinear function optimization and so on. Foraging behavior can be described as follows: let X be the current status of artificial fish swarm, and select a status Y in its reasonable range randomly. If X is superior to Y, the artificial fish moves one step towards Y. Otherwise, reselect status Y randomly and determine whether or not it meets the requirement of moving forward. If it can’t yet meet the requirement of moving forward after N attempts, move one step randomly. To deal with the situation that vacancy may appear in the neighborhood of glowworms in GSO, this foraging behavior is introduced into the iteration algorithm to ensure the position alteration of the glowworm individuals. The moving strategy is described as follows:

① If there is no better value than the current one, the current individual may be the optimal individual. Then, record the relevant information of this individual and compare it with the current optimal solution. If it is higher than the current optimal solution, make a substitute or move at the predetermined step length randomly.

② Because there may be a vacancy in the neighborhood, the foraging behavior is executed and let the maximum attempt times be N. If the movable individuals can’t be found within N search process, move randomly.

(4) Crossover strategy

To avoid premature convergence in the proposed algorithm, it is necessary to jump out of the current population, which is helpful to enhance algorithm’s global optimization ability. Suppose that $f(x)_{avg_n}$ is average fitness value of n generation and Threshold is number of comparison threshold. The crossover strategy is given as: record the fitness of best individual and average fitness of each generation firstly. If the average values haven’t change for Threshold generation, crossover operator is performed.

The method of encoding schema, crossover operator and fitness function of Hybrid-GSO are designed as follows:

(1) Encoding method

Although order scheduling problem exists sorting in group and sorting between group in sorting process. According to workpiece can be processed at any position, workpiece-based encoding method is performed because it is more intuitive and can ensure crossover offsprings are feasible solutions.

(2) Crossover operator

Linear order cross (LOC) is adopted in this paper [12]. Firstly, select a substring from the first father individual randomly and copy it to a new empty string according to the same location. Then delete the gene which appears in the first father. At last, fill in the new string’s empty locations with genes from the second father by or-
der from left to right and crossover offspring is obtained. For example, father one: \(x_1 = 1\)–2–3–4–5–6, father two: \(x_2 = 6\)–4–2–1–3–5. If genes string 3–4–5 is selected, the individual xx 3–4–5 x is found. Then \(x_2\) only left the substring 6–2–1. Fill them to the new string and we can get 6–2–3–4–5–1 as a crossover offspring. Likewise, another offspring is obtained.

(3) The fitness function

The fitness function is defined as \(F(h) = 1/(1 – O(h))\). Because \((1 – O(h))\) is a number between \([0, 1]\), the distinct changing process of \(F(h)\) appear clearly by using the reciprocal of \((1 – O(h))\).

Based on the above method, the procedure of Hybrid-GSO is as follows:

**Step 1.** Initialize parameters: Let \(l_0\) be fluorescein value. Let \(r_0\) be dynamic decision domain, \(s\) be step, \(n_t\) be threshold in domain, \(\rho\) be fluorescein elimination coefficient, \(\gamma\) be fluorescein update coefficient, \(\beta\) be updated coefficient of domain, \(r_s\) be maximal searching radius and let \(t\) be the iteration number. Then generate the initial population and calculate individuals’ fitness value;

**Step 2.** Record current best individual according to fluorescein value;

**Step 3.** Individual is chosen and move within the neighborhood among its range. Some special individuals implement foraging behavior. Update the fitness values of the current population;

**Step 4.** Calculate average fitness values of populations and judge whether or not the crossover operation is needed. If it meets crossover conditions, turn to step 5; or else to step 6;

**Step 5.** Perform crossover operator on the individuals except the best one and update values of new individuals;

**Step 6.** Update locations and dynamic decision domains of the offspring populations;

**Step 7.** If \(t < T_{\text{max}}\), go to step 2 till \(t = T_{\text{max}}\). Otherwise, to step 8.

**Step 8.** Stop iteration and record the best fitness.

### 4. Simulation Result

To verify the feasibility of Hybrid-GSO (HGSO), the following case is considered: there are a whole-set orders scheduling problem with setup time including 10 orders and 41 workpieces. Relevant information is shown in Table 1.

The proposed model is solved by classic GSO and HGSO independently. After running two algorithms 15 times with different population strategies independently, the simulation result is shown in Table 2.

From Table 2, it is known that the best solution can be found by two algorithms. However, the strategy in HGO performed better in the worst value, average and variance than the strategy in classic GSO. We can draw a conclusion that the convergence process in HGSO with

### Table 1. Order information

<table>
<thead>
<tr>
<th>Order</th>
<th>Weight</th>
<th>Processing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.08</td>
<td>1 2 2 2 2 2</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>1 2 1 2</td>
</tr>
<tr>
<td>3</td>
<td>0.09</td>
<td>1 2 1 2</td>
</tr>
<tr>
<td>4</td>
<td>0.11</td>
<td>1 2 1 1 2</td>
</tr>
<tr>
<td>5</td>
<td>0.12</td>
<td>1 1 1 2</td>
</tr>
<tr>
<td>6</td>
<td>0.2</td>
<td>1 2 2 2</td>
</tr>
<tr>
<td>7</td>
<td>0.05</td>
<td>1 2 2</td>
</tr>
<tr>
<td>8</td>
<td>0.15</td>
<td>2 1 2 1 2</td>
</tr>
<tr>
<td>9</td>
<td>0.05</td>
<td>1 1 2 1</td>
</tr>
<tr>
<td>10</td>
<td>0.05</td>
<td>2 1 2 2</td>
</tr>
<tr>
<td>Deadline</td>
<td>2 3 7 13 17 19 22 33 47 49 59 67</td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>1 1 1 2 2 2 2 2 3 3 3 3</td>
<td></td>
</tr>
<tr>
<td>Set-up time</td>
<td>5 5 5 3 3 3 3 4 4 4 4</td>
<td></td>
</tr>
</tbody>
</table>

In order to further verify the proposed algorithm’s performance, a comparison is made between the classic GSO and HGSO while both algorithms are running by same parameters. Figure 1 shows the changing curves for average fitness values of populations.

From Figure 1, the average values of populations in HGSO are better than those in GSO. In another word, individuals in HGSO have a bigger probability to meet the best solution.

Figure 2 shows that HGSO yields faster convergence than classic GSO.

Comparing with classic GSO, the searching curves in Figure 2 indicate that HGSO spends less time to find a better solution.

5. Conclusions

Whole-set orders scheduling problem with setup-time has important research significance in today’s ‘Make to order’ production mode. By analysing the characteristics of whole-set orders scheduling problem with set-up time and classic GSO, Hybrid-GSO is proposed in this paper. Foraging behavior and crossover operation are introduced in this algorithm, which is different with classic GSO and is more suitable to solve this scheduling problem. The simulation results validate the feasibility of Hybrid-GSO.

Acknowledgements

Project is supported by National Natural Science Foundation of China (No. 61573250), Youth Foundation of ShanXi Province (No. 201601D021082) and the Post-doctoral Science Fund of Taiyuan University of Science & Technology (No. 20142018).

References


