Clustering and Differential Evolution Algorithm for Solving Multi-objectives IPPS Problem

Xuan Du, Aihao Wang and Zhicheng Pan

ABSTRACT

Aiming at the multi-objective integrated process planning and scheduling problem with make span, tardiness and equipment load optimization objectives, building a multi-objective non-chain process planning integration model. Combined with the clustering algorithm, differential evolution algorithm and genetic algorithm operations, the diversity in the feasible solution space has been kept, the rapid updating of Pareto non-dominated solutions has been realized. Finally, the feasibility and superiority of the algorithm are verified by an example.

INTRODUCTION

In traditional manufacturing, because process planning and shop scheduling are independent and serial, it may led to the target conflict, unreasonable use of resources, even the emergence of material shortage of production and other phenomena in the production process, such as unreasonable order cancellation.

Based on the existing research, this paper designs a Clustering Differential Evolution Algorithm (CDE). Based on this algorithm, the optimized search not only keeps the feasible solution and the efficiency of the solution, but also improves the quality of the solution by maintaining the difference of the scheduling scheme.

Xuan Du, Aihao Wang, College of Mechanical & Power Engineering, China Three Gorges University Yichang, China, 443002
Zhicheng Pan, Yichang Changjiang Machine Technology Co., Ltd. Yichang, China, 443002
PROBLEM DESCRIPTION AND MODEL

To minimize make span \( f_1 \), total tardiness \( f_2 \), and maximum equipment load \( f_3 \), establish objective functions such as formula (1), (2), and (3):

\[
 f_1 = \min \left\{ \max \left\{ tc_i \right\} \right\} \quad (1)
\]

\[
 f_2 = \min \left\{ \sum_{i=1}^{n} \max \left\{ tc_i - D_i, 0 \right\} \right\} \quad (2)
\]

\[
 f_3 = \min \left\{ \max \left\{ \sum_{i=1}^{n} \sum_{k=1}^{p_c} \sum_{d=1}^{d_c} t_{ikd} X_{ikd} Y_{ikd} \right\} \right\} \quad (3)
\]

According to the above assumptions, the relevant constraints are as follows:
(1) The process can only be processed on one machine at the same time:

\[
 \sum_{j=1}^{d_c} Y_{ikdj} = 1 \quad (4)
\]

(2) One job can only choose a process route processing:

\[
 \sum_{k=1}^{p_c} X_{ik} = 1 \quad (5)
\]

(3) The job must be processed in sequence according to its process route:

\[
 tc_{ikd_j} X_{ik} Y_{ikd_j} - tc_{ik(d-1)j} X_{ik} Y_{ik(d-1)j} \geq t_{ik(d-1)j} X_{ik} Y_{ik(d-1)j}, \quad j_1, j_2 = 1, 2, \ldots, m \quad (6)
\]

(4) A device at the same time can only process a process:

\[
 ts_{ikd_{1z}} X_{ik} Y_{ikd_{1z}} - ts_{ikd_{2z}} X_{ik} Y_{ikd_{2z}} \geq t_{ikd_{2z}} X_{ik} Y_{ikd_{2z}}, \quad d_1, d_2 = 1, 2, \ldots, p \quad (7)
\]

ALGORITHM DESIGN AND ANALYSIS

Encoding and Decoding

<table>
<thead>
<tr>
<th>Table I. Examples of Codes for Feasible Solutions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job type</td>
</tr>
<tr>
<td>Process chain</td>
</tr>
<tr>
<td>Equipment chain</td>
</tr>
<tr>
<td>Processing sequence chain</td>
</tr>
</tbody>
</table>

251
In this paper, a three-layer encoding structure is designed based on the job type including the process chain, equipment chain and processing sequence chain, respectively, that the job process routes, the processing of the equipment and order of processing. TABLE I shows the example of encoding.

### Crossover Operator and Mutation Operator

Find feasible solutions with more dominating numbers and select the feasible solution with the largest Haiming distance from the current processing sequence chain, use single point crossover to update their processing sequence chain. The Figure 1 shows the crossover process.

In this paper, a fast non-dominated sorting method is adopted[1]. Select the feasible solution with less dominated number to do the search. In the discrete multi-objective problem, when searching, the amount of non-dominated solution is less. So to solve this problem, after each iteration, store the top m of feasible solution in the non-dominated solution sequence. However, because some solutions are in the same layer. So calculate the crowding degree [2] to choose a feasible solution.

### INSTANCE ANALYSIS

#### Instance Information

This paper based on MATLAB software platform programming CDE algorithm to solve multi-target IPPS problem, 20×5[3] question is tested respectively (a×b indicates that a kinds of jobs are machined on b kinds of machines) to test the performance of CDE algorithm, and comparing with the existing results, the efficiency of the algorithm is tested. TABLE II is algorithm related parameters.
Effectiveness Analysis

Table II. The 20×5 Test Result Table.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>GRASP</th>
<th>HBMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>924</td>
<td>708</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>2919</td>
<td>1780</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>889</td>
<td>708</td>
</tr>
</tbody>
</table>

Rajkumar[3] and WEN[4] proposed the GRASP algorithm and HBMO algorithm to optimize the 20×5 problems. In this paper, compare the CDE algorithm with the results of the above algorithm to verify CDE algorithm effectiveness, the results are as Table II.

Table II shows Pareto optimal solution sets for solving different 20×5 multi-objective IPPS problems with different algorithms. As can be seen, compared to HBMO algorithm, CDE algorithm improves the Pareto optimal solution set in the number and quality significantly. The optimal value of each target are 2.54%, 35.88% and 5.93% better respectively compared with HBMO algorithm.

CONCLUSIONS

Using GA and DE algorithms to update feasible solution purposely, keeping the solution set in the search space well distributed, avoiding to fall into a local optimum. After that, the clustering algorithm is used to update the local optimal solution, making the solution set quickly jumps out of the local optimum which not only improves the efficiency of the solution, but also improves the quality of the population.

REFERENCES