Integrated scheduling algorithm for multiple complex products with due date constraints

Xinkun Wang 1,a, Yuchuan Song 2,b*, Yuji Zou 3,c, Weifei Guo 4,d, Yi Wang 5,e

College of Mechanical Engineering ChongQing University, ChongQing University, ChongQing

*aemail: 20180702005t@cqu.edu.cn, *email: 201807021067@cqu.edu.cn,
*bemail: 20140713232@cqu.edu.cn, *email: 20180702002t@cqu.edu.cn

*Corresponding author: *email: syc@cqu.edu.cn

Abstract: A new scheduling algorithm based on improved genetic algorithm is presented to solve integrated scheduling problems in small batches of multiple complex products with different manufacturing and due dates. This algorithm virtualizes the schedule of multiple complex products into a single scheduling of one product by establishing a relationship matrix of operation of multiple products to facilitate complete optimization. In addition, novel coding, crossover, mutation, and encoding are designed for the improved genetic algorithm. The proposed integrated scheduling algorithm is tested by existing examples. Experimental results validate the effectiveness and superiority of the proposed integrated scheduling algorithm.

1. Introduction

Production scheduling is an important combinatorial optimization problem. The computational complexity of most production scheduling problems is NP-hard, and it is difficult to optimize the solution [1]. Tardiness scheduling problems were first proposed by Iwata et al. in 1980. Specifically, they applied the concept of earliest finish time to solve the problem [3-8]. However, these algorithms mainly solve problems of pure processing or pure assembly and are suitable for batch production.

Orders for small batches of multiple complex products continue to increase as a result of increasing demand for product diversity. Traditional job-shop scheduling algorithms inevitably and increasingly split the inherent parallel processing relationship between machining and assembly, thereby affecting the production and economic efficiency of an enterprise. Integrated scheduling of complex products is proposed to solve this problem.

This study adopts an improved genetic algorithm based on the integrated and whole thoughts of virtualizing the schedule of multiple products into a single schedule of one product to solve the integrated scheduling problem of small batches of multiple complex products with different manufacturing and due dates. The mathematical model of the problem and the relationship matrix of operation of multiple products are established to realize overall scheduling and subsequent calculations. This study also aims to address situations where existing coding methods and evolution operators cannot be applied in integrated scheduling. Novel coding, crossover, mutation, and encoding methods are designed for the improved genetic algorithm. The effectiveness and superiority of the proposed algorithm are verified by examples.
2. Formulation of the Problem

Compared with general integrated scheduling problems, this study considers the manufacturing and due dates of small batches of complex multiple products and the studied problem becomes much more complicated.

To facilitate description, we assume a set of $w$ products $\{P_i\}_{i=1}^{w}$ that include $n$ operations that will be processed on $m$ machine $\{M_k\}_{k=1}^{m}$. Product $P_i$ is formed by $n_i$ operations $\{o_{ij}\}_{j=1}^{n_i}$. $S_i$, $C_i$ and $D_i$ are manufacturing date, the completion time and the due date of the product $P_i$, respectively. $s_{ijk}$, $t_{ijk}$ and $c_{ijk}$ are start time, the processing time, and finish time of the operation $o_{ij}$ on machine $M_k$.

Operation $o_{iq}$ is the immediate predecessor operation of operation $o_{ij}$. $X_{ijeg} = 1$ denotes that operations $o_{ij}$ and $o_{eg}$ are processed on the same machine and the former is prioritized. $Y_{ijk} = 1$ denotes that operation $o_{ij}$ is processed on machine $M_k$. $f$ denotes the optimization objective of primary scheduling and $T$ denotes the secondary objective. The relationship of the two objectives: Under the premise of meeting the Objective (1), the scheduling scheme that meets the Objective (2) is selected as the optimal scheduling scheme.

The two objectives can be written as:

Primary objective: 
$$ f = \min(\sum_{i=1}^{w}\{\max(C_i - D_i, 0)\}) $$  \hspace{1cm} (1)

Secondary objective: 
$$ T = \min(\max C_i) $$  \hspace{1cm} (2)

Subject to the following constraints:

Constraint (3) The start time of each product operation can only be after the manufacturing date of the product.

Constraint (4) The finish time of each product operation must be in the due date of the product.

Constraint (5) Any product operation can begin after its immediate predecessor operations have finished processing or no immediate predecessor operations occur.

Constraint (6) One machine can only process one operation of one product at most at any given time.

3. Algorithm Design

This study presents a series of operating methods of a genetic algorithm to solve the integrated scheduling problem of small batches of multiple complex products.

3.1 Establishing the Relationship Matrix of Operation of Multiple Products

In this study, 0–1 programming is used to describe the characteristics of timing constraints among operations and to establish the relationship matrix of operation of multiple products, which not only express complex timing constraints among operations in multiple products, but also realize multiple products scheduling and assist subsequent genetic operations. This method is defined as follows.

If operation $O_{iq}$ is the immediate predecessor operation of operation $O_{ij}$, element value $V_{ij_{ijq}}$ is 1; otherwise, it is 0 and element value $V_{ij_{ijj}}$ is 0 in the relationship matrix of operation of multiple products. A relationship matrix of operation of multiple products can be established based on the preceding definitions. The directed graph of the operation relationships is shown in Figure1. The corresponding relationship matrix of operation of the products is provided in Table 1.
3.2 Coding Based on the Dynamic Relationship Matrix of Operation of Multiple Products

Genetic coding is the most important problem in the process of the optimization of genetic algorithm. Given the advantages of decoding by simple mapping, every operation involving multiple products must be considered a single-operation job. This approach can reduce computation and enables operations to be divided into schedulable and non-schedulable operation sets regardless of the machine conditions. The steps are described as follows.

Step 1: Obtain the schedulable operation set $S$ by identifying the relationship matrix of operation of multiple products. If $v_{ijq}$ is 0, then $o_q$ has no constraint on $o_j$. Thus, if the values of $v_{ijq}$ are all 0 at one time, then $o_q$ is a schedulable operation. This approach provides a schedulable operation set $S$.

Step 2: Choose an operation $o_q$ that will be optionally processed from schedulable operation set $S$ and update the relationship matrix of operation of multiple products by deleting all values of $v_{ijq}$ to relieve the constraints of operation $o_q$ on other operations and the constraints of others on it.

Step 3: Repeat Step 1 and determine schedulable operation set $S$. The procedure is terminated if $S$ is empty; otherwise, proceed to Step 2.

A chromosome can be obtained and the timing constraints of operations of multiple products can be satisfied based on these steps. Figure 2 presents a schematic diagram of chromosome coding of products A, B, and C. Figure 2 shows that 1–11 correspond to operations A1–A11 of product A; 12–19 correspond to operations B1–B8 of the first product B; 20–27 correspond to operations b1–b8 of the second product B; and 28–38 correspond to operations C1–C11 of product C.

3.3 Selection Operator

Selection operator aims to promote global convergence and improve computation efficiency. The proposed algorithm adopts the most basic and common roulette selection method.
### Table 1. Relationship matrix of operation of products A, B and C

| Operation | A | A | A | A | A | A | A | A | B | B | B | B | B | B | B | C | C | C | C | C | C | C | C | C | C |
|-----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A2        | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A3        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A4        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A5        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A6        | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A7        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A8        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A9        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A10       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A11       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B1        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B2        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B3        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B4        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B5        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B6        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B7        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B8        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B9        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B10       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B11       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C1        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C2        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

3.4 Crossover Operator

Crossover operator is the most important operation because it determines global searching ability. We design single-point crossover and two-point crossover methods for the integrated scheduling problem.

**Single-point crossover method:** Randomly selected a cross position in two-parent chromosomes, C1 (C2) retains the left end gene of the cross position of the parent P1 (P2), and the right end gene of the cross position of the parent P2 (P1). Figure 3 shows the procedure for product A in Figure 1.

**Two-point crossover method:** Randomly select two different crossover sites in the parental chromosomes. C1 (C2) retains the leftmost and rightmost genes of the parent P1 (P2), and the unselected gene sequence of the parent P2 (P1) is placed between the two crossover sites from left to right. Figure 4 shows the procedure for product A in Figure 1.

3.5. Mutation Operator

The mutation operator changes the value of some genes in the chromosome with a certain probability of maintaining the diversity of the population. We design two new mutation methods.

3.5.1. Backward mutation
First, randomly select a variant from one parent chromosome $P$. Second, delete all values of corresponding rows and columns of genes before the mutation site. Third, determine the number of schedulable operations. Finally, a different schedulable operation on the mutation site is selected and other genes are produced by generating the initial chromosome. Figure 5 shows the procedure.

3.5.2. Forward mutation
First, randomly select from one parent chromosome $P$. Second, delete the values of the corresponding rows and columns of genes after mutation. Third, determine the number of schedulable operations. Finally, a different schedulable operation on the mutation site is selected and other genes are produced by generating the initial chromosome. The procedure for product A is shown in Figure 6.

3.5.3. Analysis of the mutation methods
The two types of mutation will not produce infeasible individuals. An individual generated through any of the aforementioned mutation methods may not be unique under the same mutation position.

3.6. Decoding Based on the Relationship Matrix of Operation of Multiple Products
The genetic algorithm adopted an active decoding mechanism which was better than the semi-active mechanism in the quality of solution. Thus, the proposed algorithm adopts the insert greedy decoding algorithm. The basic procedures are described as follows.

Step 1: The start time of each job (each operation) is set to the corresponding manufacturing date of its job during the initial state.

Step 2: A gene is read from left to right from the coding chromosome. Greedy decoding is performed and the completion time of its corresponding operation $i_{oj}$ is recorded.

Step 3: Operation set $C$ constrained by operation $i_{oj}$ according to the relationship matrix of operation of multiple products is obtained. The start time of each operation in set $C$ should be comparable with the completion time of $i_{oj}$. The longer time is then used as the new start time of the corresponding operation in operation set $C$. The related data of the corresponding machines are simultaneously updated.

Step 4: If the product is complete, the process is terminated; otherwise, proceed to Step 2.

3.7. Flowchart of the Proposed Algorithm
The flowchart of the proposed algorithm is shown in Figure 7.
4. Sample Comparison and Analysis

The proposed algorithm is coded with the following parameters: size of the population \(\text{pop\_size}\) is 200, total generations \(\text{max\_gen}\) is 50, probability of crossover operator \(p_c\) is 0.8, and probability of mutation operator \(p_m\) is 0.1. The proposed algorithm is run 10 times for each solution in the sample. Figure 1 illustrates the directed graph of operation relationship of products A, B, and C in the sample. Table 1 shows their corresponding operation relationship matrix.

**Solution 1:** The manufacturing dates of products A, B and C are 0, whereas their respective due dates are 250, 550 and 500. The Gantt chart obtained using integrated scheduling algorithm based on urgency of delivery dates is shown in Figure 8. The total processing time is 465. Figure 9 shows the Gantt chart obtained using the algorithm. The total processing time is 430. These finding indicate that the scheduling result of the proposed algorithm meets the due dates with reduced processing time.

**Solution 2:** The manufacturing dates of products A, B, and C are 0, 130 and 130, and their due dates are 250, 550 and 500. The Gantt chart obtained using the integrated scheduling algorithm based on urgency of delivery dates is shown in Figure 10. The total processing time is 480. Figure 11 shows the Gantt chart obtained using the algorithm in this study. The total processing time is 435. These
results show that the proposed algorithm meets the due dates and reduces total processing time.

**Solution 3:** The manufacturing dates of products A, B and C are 0, 130 and 130, and the respective due dates are 250, 476 and 400. The Gantt chart obtained using the integrated scheduling algorithm based on urgency of delivery dates is shown in Figure 12. The total processing time is 475. The Gantt chart obtained using the algorithm is shown in Figure 13. The total processing time is 465. These results show that the proposed algorithm meets the due dates with reduced total processing time.

5. Conclusions

A novel integrated scheduling algorithm based on improved genetic algorithm was proposed to solve integrated scheduling problems for small batches of complex multiple products with different manufacturing and due dates. The main innovations are as follows: (1) Use a relationship matrix for operation of multiple products and treat them as a single virtual product, which overcome the deficiencies of the existing algorithm through partial integral solution; (2) Establish a mathematical model with the primary goal of meeting due dates and the secondary goal of minimizing total processing time to ensure the quality of final scheduling; (3) Design the corresponding novel coding, crossover, mutation and encoding methods of the improved genetic algorithm to solve the problem. This approach can serve as a reference for other intelligent algorithms when solving similar problems.

The proposed algorithm can overcome the deficiencies of the existing algorithm and can solve large-scale integrated scheduling problems. The new algorithm is simple, practical and can provide reference values for further research on multiple complex products scheduling problems with due date constraints.

**Acknowledgments:**

This article is one of the initial results of “the National Natural Science Foundation of China”
References:

[1] Ahmadizar, Fardin; Hosseini, Leila. (2011). A novel ant colony algorithm for the single-machine total weighted tardiness problem with sequence dependent setup times. International Journal of Computational Intelligence Systems, 4(4), 456-466.

[2] Yi Zhang, Junlong Zhou, Jin Sun. (2019). Scheduling bag-of-tasks applications on hybrid clouds under due date constraints, 101.

[3] Kuhpfahl, J; Bierwirth, C. (2016) A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective. Computers & Operations Research, 66, 44-57.

[4] Ghassemi Tari, Farhad; Olfat, Laya. (2014). Heuristic rules for tardiness problem in flow shop with intermediate due dates. International Journal of Advanced Manufacturing Technology 71(1-4), 381-393.

[5] Schaller, Jeffrey. (2012). Scheduling a permutation flow shop with family setups to minimise total tardiness. International Journal of Production Research, 50(8), 2204-2217.

[6] Y. Kergosien, M. Gendreau, J.-C. Billaut. (2017). A Benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints, 262(1):287-298.

[7] Xie, Zhiqiang; Wang, Peng; Gui, Zhongyan; Yang, Jing. (2012). Integrated scheduling algorithm based on dynamic essential short path. Advances in Intelligent and Soft Computing, 169(2), 709-715.

[8] Xie, Zhiqiang; Chang, Ningning; Yang, Jing. (2011). Integrated scheduling algorithm based on delivery dates urgency degree. Chinese Journal of Mechanical Engineering, 47(22), 181-190.