International Journal of Industrial Engineering Computations 14 (2023) 749–766

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Collaborative scheduling of machining-assembly in complex multiple parallel production lines environment considering kitting constraints

Guangyan Xu[a], Zailin Guan[a], Kai Peng[b] and Lei Yue[b*]

[a]*School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, P.R. China*
[b]*School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou, P. R. China*

C H R O N I C L E

Article history:
Received March 1 2023
Received in Revised Format May 10 2023
Accepted July 28 2023
Available online July, 28 2023

Keywords:
Fabrication -assembly
Collaborative scheduling
Multiple objectives
Complete set
MOEA/D

A B S T R A C T

In multi-stage machining-assembly production, collaborative scheduling for multiple production lines can effectively improve the execution efficiency of production planning and increase the effective output of the production system. In this paper, a production scheduling mathematical model was constructed for the collaborative scheduling problem of machining-assembly multi-production lines with kitting constraints, with the optimization objectives of minimizing assembly completion time and tardiness time. For the scheduling model, the product assembly process is constrained by the machining sequence of the jobs on the machining lines. Only by collaborating on the production scheduling schemes of the machine line and the assembly line as a whole can the output efficiency of the product on the assembly line be improved. An improved hybrid multi-objective optimization algorithm named SMOEA/D is designed to solve this scheduling model. The algorithm uses adaptive parents' selection and mutation rate strategies and integrates the Tabu search strategy for the search process in the solution space when the solution of the sub-problem has not been improved after specified search generations, to improve the local search ability and search accuracy of MOEA/D algorithm. To verify the performance of the SMOEA/D algorithm in solving machining-assembly collaborative scheduling problems in production systems with different resource configurations and scales, two sets of numerical experiments were designed, corresponding to situations where the number of operations on each production line is equal or unequal. The running results of the proposed algorithm were compared with three other well-known multi-objective algorithms. The comparison results indicate that the SMOEA/D algorithm is effective and superior for solving such problems.

1. Introduction

Driven by market demand and technological innovation, enterprises often need to produce various types of products under limited resources to meet the diverse and personalized needs of customers. In order to produce multiple types of products under limited resources of the production system, the mixed-model processing and assembly is widely adopted by enterprises. Especially in the production of complex industrial finished products, such as automobiles, engineering machinery, and home appliance industries, enterprises often only produce their core components, while other non-core and universal components are generally obtained through external procurement through the supply chains, and then the assembly process of Multi-variety products is completed on the assembly line. In this production environment, the planning execution status and scheduling scheme of the core component directly affect the output efficiency of the assembly production line. In the manufacturing workshop of the coach drive axle in a certain enterprise, due to the needs of platform-based and serialized production of coach products, various components of different models of the driving axle are assembled and produced on the assembly line. As shown in Fig. 1, there are many components used in the assembly of the driving axle. In addition to external

* Corresponding author
E-mail: leileiyok@gzhu.edu.cn (L. Yue)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2023 Growing Science Ltd.
doi: 10.5267/j.ijiec.2023.7.003

parts such as standard parts and universal parts, some key components of the driving axle, such as the axle housing, main reducer housing, differential housing, and bearing housing, are produced by the enterprise's own organized production. After completing machining production on the corresponding machining line, these components, along with the purchased parts, undergo the assembly and production process of the driving axle products on the assembly line. Due to the involvement of multiple varieties of production, the assembly of the driving axle on the assembly line must meet the kitting constraints. Any shortage of subcomponents will cause assembly production to be unable to start, which means that the driving axle assembly production is constrained by the scheduling scheme and type of subcomponents. In this production context, only by coordinating the scheduling scheme of the driving axle assembly and its subcomponents based on the relative optimization level of demand orders can the assembly line's kitting be improved, thereby improving its effective output. This study is carried out in this context.

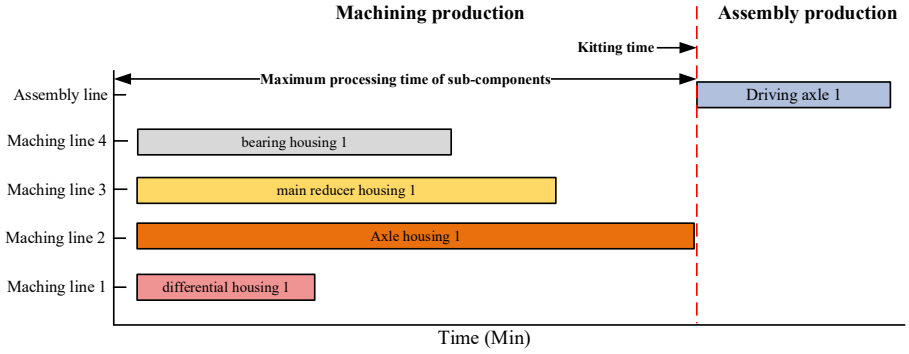

Fig. 1 Schematic diagram of machining-assembly production

The collaborative scheduling problem of processing-assembly of multiple production lines is very common in manufacturing enterprises, where many industrial finished products are produced by manufacturing or purchasing components, and then the final product is produced through the assembly lines. And these components are often processed in different enterprises, workshops, or production lines according to the needs of the final product, and then assembled on the final product assembly line. This production mode is more typical in automotive manufacturing. For multi variety mixed flow assembly production, collaborative scheduling in the multiple stages of machining assembly can improve the kitting during assembly production, reduce inventory backlog of middleware, promote the fluidity of the entire production process, and improve the output efficiency of assembly production. Therefore, the research content of this article has important practical value and theoretical significance.

The remainder of this paper is organized as follows: Section 2 provides an overview of the current research status of collaborative scheduling problems; Section 3 formulates a mathematical model for the scheduling problem based on the characteristics of multi variety machining-assembly production. Section 4 proposes a new hybrid multi-objective optimization method SMOEA/D for solving such problems. Section 5 designed numerical experiments and analyzed the laboratory results. Section 6 provides conclusions and prospects for future research work.

2. Literature reviews

The refinement of production processes and the deep application of information technology in the manufacturing industry provide a driving force and technical foundation for the research of collaborative scheduling problems. By considering multiple stages in the production system or supply chain as a whole, the connectivity between each stage is improved, thereby improving the quality and execution efficiency of scheduling schemes. Due to the widespread use of collaborative scheduling in industries such as workshop production, logistics transportation, and aerospace, scholars have increased their research in this field in recent years and achieved a series of research results.

Meng et al. (2017) studied the two-stage scheduling problem of structural component machining-welding. In the established problem model, the start time added in the second stage is constrained by the completion time in the first stage. By considering the dominant relationship between the structural component and its corresponding subcomponents, the total completion time is minimized. In the proposed improved harmonious search algorithm, a local search method is used to locally search for the best vector during each iteration to achieve more good vectors. Numerical experiments have shown the effectiveness of the proposed algorithm. Sun et al. (2023) established a multi-resource collaborative scheduling optimization model based on the research object of the multi-resource collaborative scheduling problem, whose goal is to minimize the completion time and transportation energy consumption of quay cranes. In order to solve this kind of problem, they designed a hybrid algorithm SA-GA algorithm that integrates simulated annealing algorithm into genetic algorithm to solve the mixed Integer programming model. The effectiveness and superiority of the SA-GA algorithm for problem models have been demonstrated through numerical experiments. Chao et al. (2018) and Lu et al. (2017) studied the multi-objective optimization of multi machine collaborative welding workshop scheduling, which can significantly reduce production completion time and energy consumption by optimizing task scheduling within limited resources. Yang et al. (2016) proposed a flowshop scheduling

problem for multiple production lines in prefabricated production, established a Flow Shop Scheduling Model (MP-FSM) for multiple production lines, and used GA algorithm to solve the MP-FSM problem. The experimental results show that using this method can obtain optimized scheduling schemes. Literature (Deng & Wang, 2017; Kuo et al., 2007; Pan et al., 2011; Wang et al., 2011; Yue et al., 2019) analyzed the collaborative scheduling problem of production and transportation based on batch processing. Hall and Potts (2003); (Hall & Potts, 2005) analyzed the coordinated scheduling problem of single or parallel machine production and batch delivery in a supply chain environment. Ullrich and Christian (2013) studied a collaborative scheduling problem with a customer time window, with the goal of minimizing the total delivery time delay, and designed a genetic algorithm to solve it. Bhatnagar et al. (1993) summarized and analyzed the collaborative scheduling problem of vertically integrated multi workshops, and constructed a mathematical model for multi workshop collaboration to ensure that product production and inventory decisions reach an overall optimization among multiple workshops. Mazdeh et al. (2008) optimized the production and transportation collaborative scheduling problem with batch transportation with the goal of minimizing the total process time and delivery cost of chemical components, and proposed a branch and bound algorithm to solve the problem. Behnamian and Ghomi (2016) organized and classified literature on multi workshop scheduling based on the workshop environment, quantitatively compared the reviewed literature, and raised some issues that received less attention.

From the above literature review, the research on collaborative scheduling problems in the reviewed literature focuses on collaboration between upstream and downstream of the supply chain and multiple production units with the same function. However, research on collaborative scheduling of machining-assembly multiple production lines is rare. In the research reviewed, there are few literatures that studies the machining and assembly processes as a whole; Instead, they are modeled as two different scheduling problems, or assembly production is studied as a special machining process, ignoring or reducing the influence of assembly on component machining. In the assembly process of a product, it usually involves multiple subcomponents. Therefore, when studying collaborative scheduling, it is often necessary to consider the collaboration between the assembly process and multiple machining production processes. For mixed-model assembly production, the assembly process is constrained by kitting, so these problems are typical. At present, the research on this problem in literature is not in-depth, and there are few research results involving collaborative scheduling of machining-assembly production. For multi-product collaborative scheduling problems of machining-assembly, it is usually necessary to consider the kitting constraints of subcomponents and products, which requires overall co-scheduling of key production lines to achieve the global optimal scheduling effect of the entire production process.

3. Problem Description and Mathematical modeling

This section will establish a corresponding mathematical model considering kitting constraints according to the characteristics of the machining-assembly production system. The general description of the collaborative scheduling problem model for machining-assembly multiple production lines is as follows: there are T different types of product for assembly production on assembly line L. Any product is composed of n_i key components, and subcomponents i' ($i' \in \{1,2,\cdots,\}$) have T'types and are processed in a mixed flow manner on machining production line l', and the available number of machines on stage k of machining production line l' is $M_{l',k}$,$M_{l',k} \geq 1$. On assembly production L, n' subcomponents are re-pulled to complete their production on the machining production line according to the requirements of the assembly line. At any time, there can only be one job or product on a machine (or assembly station) for processing or assembly. The assembly process of any product on assembly line L is limited by the kitting constraints of subcomponents on each machining line, which means that the assembly process of any product can only begin after n' subcomponents required for this product have been processed.

Next, a mathematical model for collaborative scheduling of machining-assembly will be established. To increase the universality of the scheduling model, the problem has the following assumptions:

(1) At the initial moment, all jobs on the machining production line can be processed;
(2) Ignore the transportation time and quality inspection time of all jobs;
(3) The production preparation time of the job is included in the processing time;
(4) Neglecting the disturbance caused by external emergencies to production;
(5) Once any process of the job is started on a certain machine, it cannot be interrupted until the process is completed;
(6) The processing time of each operation is fixed;
(7) All machines are available at the initial moment, without considering the occurrence of machine failure;
(8) Not considering quality issues and rework;
(9) Do not consider setup time.

The symbols and related explanations to be used in this model are shown in the table below:

Symbol	Description
l	Production line index;
i, i'	Job index；
s	Operation index;
k	Machine Index；
N_l	The number of jobs on production line l;

L	The number of production lines, When $l = L$, l is assembly line; Otherwise, l is the machining lines;
S_l	The number of stages in production line l, also equal to the number of operations of production line l;
$M_{l,s}$	Number of machines in stage s of production line l;
$ST_{l,i,s}$	The start time of operation s job i on production line l;
$CT_{l,i,s}$	The completion time of operation s job i on production line l;
$CT_{l,i}$	The completion time of job i on production line l;
$PT_{l,i,s}$	The processing time of operation s of job i on production line l;
$PT_{l,i,s,k}$	The processing time of operation s of job $i$on machine k of production line l, t
$D_{L,i}$	The due date of product i on the assembly line;
M	A sufficiently large positive number;
$AST_{l,s}$	The earliest available time of production line$l$
$Tard_{l,i}$	The tardiness time of job i on production line l.

The definition of decision variables is as follows.

Notation	description
$X_{l,i,i',k}$	Binary variable, when job i is processed before job i' on machine k of production line l, this variable is taken as 1; Otherwise, take 0.
$Y_{l,i,s,k}$	Binary variables, when the operation s of job i is processed on machine k of production line l, this variable is taken 1; Otherwise, take 0.

Establish a mathematical model for the collaborative scheduling problem of machining-assembly multi-production line using the above symbols. The mathematical model aims to minimize the maximum completion time and the total tardiness time, and the calculation methods for these two objectives are shown in formulas (1) and (2), respectively.

$$min\ f_1 = max\{CT_{L,i}\} \tag{1}$$

$$min\ f_2 = \sum_{i=1}^{N_L} Tard_{L,i} \tag{2}$$

where $CT_{L,i}$ represents the assembly completion time of product i, $Tard_{L,i}$is the delay time of product i, and its corresponding calculation method is as follows:

$$Tard_{L,i} = \begin{cases} CT_{L,i} - D_{L,i}, & CT_{L,i} \geq D_{L,i} \\ 0, & CT_{L,i} < D_{L,i} \end{cases} \tag{3}$$

The constraints of the mathematical model for this scheduling problem are as follows.

$$\sum_{k=1}^{M_{l,k}} Y_{l,i,s,k} = 1, \quad \forall l = 1,2,\cdots,L\ \ \forall i = 1,2,\cdots,N_l\ \ \forall s = 1,2,\cdots S_l \tag{4}$$

$$\sum_{i=1}^{N_l} Y_{l,i,s,k} \leq 1, \quad \forall l = 1,2,\cdots,L\ \ \forall s = 1,2,\cdots,S_l\ \ \forall k = 1,2,\cdots,M_{l,s} \tag{5}$$

$$ST_{l,i,s} + \sum_{k=1}^{K_l} PT_{l,i,s} Y_{l,i,s,k} \leq ST_{l,i,(s+1)}\ \forall\, l = 1,2,\cdots,L\ \forall i = 1,2,\cdots,N_l\ \forall s = 1,2,\cdots,S_l\ \ k\ \ = 1,2,\cdots,M_l \tag{6}$$

$$CT_{l,i,s} = ST_{l,i,s} + PT_{l,i,s}\ \ \forall l = 1,2,\cdots,L\ \ \forall i = 1,2,\cdots,N_l\ \ \forall s = 1,2,\cdots,S_l \tag{7}$$

$$ST_{l,i,k} + PT_{l,i,k} \leq ST_{l,i',k} + M\left(1 - X_{l,i,i',k}\right)\ \ \forall l = 1,2,\cdots,L\ \forall i, i' = 1,2,\cdots,N_l \quad k = 1,2,\cdots,M_{l,s} \tag{8}$$

$$ST_{l,i,s} \geq ST_{l,i,(s-1)} + \sum_{k=i}^{M_s} \left(PT_{l,i,s,k} Y_{l,i,s,k}\right);\ \forall l = 1,2,\cdots,L\ \ \forall i = 1,2,\cdots,N_l\ \ s = 1,2,\cdots\cdots,S_l \tag{9}$$

$$ST_{l,1,1} = 0 \ \ \forall l = 1,2,\cdots,L \tag{10}$$

$$ST_{l,1,s} = ST_{l,1,(s-1)} + PT_{l,1,(s-1)} \ \ \forall l = 1,2,\cdots,L, \qquad s = 2,3,\cdots,S_l \tag{11}$$

$$ST_{l,i,s} = max\left\{CT_{l,i,(s-1)}, \sum_{k=1}^{M_{l,s}} PT_{l,i,s}Y_{l,i,s,k}\right\} \ \ \forall l = 1,2,\cdots,L, i = 1,2,\cdots,N_l \ \ s = 2,3,\cdots,S_l \tag{12}$$

$$CT_{l,i,s} = ST_{l,i,s} + \sum_{k=1}^{M_{l,s}} \left(PT_{l,i,s}Y_{l,i,s,k}\right) \ \ \forall l = 1,2,\cdots,L \ \forall i = 1,2,\cdots,N_l \ \ s = 1,2,\cdots\cdots,S_l \tag{13}$$

$$ST_{L,1,1} \geq max\left\{CT_{1,1}, CT_{2,1}, \cdots, CT_{(L-1),1}\right\} \tag{14}$$

$$ST_{L,i,1} \geq max\left\{CT_{1,i_1}, CT_{2,i_2}, \cdots, CT_{(L-1),i_{(L-1)}}, CT_{L,(i-1)}\right\} \tag{15}$$

Constraints (4) ensure any operation of the job can only be processed on one machine in any production line. Constraints (5) restrict any machine to process only one job at a time. Constraints (6) specify the start time of any operation of each job. Constraints (7) define the relationship between start time and completion time. Constraints (8) guarantee the start time constraints between the job processed by one machine and the subsequent job. Constraints (9) impose the start time of any operation of the job shall not be earlier than the completion time of its previous operation. Constraints (10)-(13) define the equation for the completion time of any operation of the jobs on each production line. When products are assembled on the assembly line, the types of subcomponents used in different products may different. Therefore, in the machining-assembly production system, the assembly process on the assembly line L is constrained by kitting of the subcomponents, that is, the assembly start time of any product must be later than the completion time of all its subcomponents used. Constraints (14) and (15c) define the kitting constraints for the production system.

4. Hybrid SMOEA/D Algorithm Based on MOEA/D and tabu Search Strategies

This section briefly introduces MOEA/D and Tabu search, and then integrates adaptive individual selection strategy and Tabu search into MOEA/D. The SMOEA/D algorithm was proposed.

4.1 Introduction to MOEA/D and tabu search

Based on decomposition algorithms such as MOEA/D (Li & Zhang, 2008), NSGAIII (Deb & Jain, 2014), and SPEAR (Jiang & Yang, 2017), among which the most famous and widely used is MOEA/D. This algorithm was first proposed by Zhang et al. in 2007 and is the most representative decomposition based multi-objective optimization algorithm. Unlike classic multi-objective optimization algorithms such as SPEA2 and NSGAII, MOEA/D decomposes a multi-objective optimization problem (MOP) into a series of single objective optimization subproblems using a set of uniformly distributed weight vectors, and simultaneously optimizes multiple single objective optimization subproblems using co evolution mechanism in one generation, Simple and efficient. A key component of MOEA/D is the decomposition method. Different decomposition methods have different advantages and shortcomings, which can affect the optimization performance of the algorithm. The commonly used aggregation function includes the weight sum approach, Chebyshev approach and the normal boundary intersection (BNI) approach. The weight method can better solve the convex MOPs problem, and The Chebyshev approach can handle non-convex MOPs, and they are all sensitive to the scale of optimization objectives (Zhang et al., 2010). In this paper, the Chebyshev method is used to decompose multi-objective optimization problems into single objective scalar optimization subproblems. Tabu search (TS) (Hao et al., 2013) algorithm is an expansion of local neighborhood search and a global gradual optimization algorithm. It avoids circuitous search by introducing a flexible storage structure and corresponding tabu criteria, and pardons some tabu good states through aspiration criteria, thus ensuring effective exploration of diversification to finally achieve global optimization. The TS algorithm has strong local search ability, but its global search ability is limited, making it easy to fall into local optima.

In this paper, to better solve the collaborative scheduling problem of machining-assembly multi production lines, we integrate the search strategy of TS into the traditional MOEA/D, and propose an enhanced MOEA/D algorithm that integrates TS search strategy, which is abbreviated as SMOEA/D. This algorithm has made some improvements in the initial population generation method, parent individual selection strategy, individual update method, and local search strategy to further balance the global and local search capabilities of the algorithm and improve the solution quality and convergence speed of the MOEA/D algorithm.

4.2 SMOEA/D algorithm framework

The basic idea of the SMOEA/D hybrid algorithm is as follows: when optimizing subproblems in each generation, genetic operators are used to generate new solutions. if the quality of the subproblems is not improved, the corresponding mutation rate of the subproblem is increases to avoid falling into local optima; When the solution of the subproblem is not improved after the specified number of generations, the TS algorithm is used to perform the Tabu search process with the current optimal solution as the initial solution. By combining the global search, adaptive strategy, and TS local search capabilities of the MOEA/D algorithm in this way, its advantages are fully utilized to improve the algorithm's solving efficiency and search accuracy. The following will provide the algorithm process and provide a detailed explanation of the main steps.

The following will provide the execution process of the MOEA/D framework in Algorithm 1.

Algorithm 1: The framework of SMOEA/D

Input data:

A multi-objective problem (MOP) to be solved;

A stopping condition;

N: The population size, equal to the number of weight vectors;

T: The neighborhood size of each subproblem;

l: Evaluation threshold for performing Tabu search;

Output data:

External Archive Set (EP).

Step 1) Initialization:

Step 1.1) Set EP=∅, s=0, s is Tabu search count;

A set of uniformly distributed weight vectors:$\lambda^1, \lambda^2, \cdots, \lambda^N$;

Step 1) Initialization:

Step 1.1) Set EP $= \emptyset, s = 0, n_i = 0, i = 1,2, \cdots, N$, n_i is the counter for the number of evaluations that the solution of subproblem i has not been improved;

Step 1.2) Generate a set of uniformly distributed weight vectors , $\lambda_1, \lambda_2, \cdots, \lambda_N$, calculate the Euclidean distance between any two weight vectors, the nearest T weight vectors of λ^i are denoted as $\mathrm{B}(i) = \{i_1, i_2, \cdots, i_T\}$, $\forall i = 1, 2, \cdots, N$, and labeled $\mathrm{B}(i)$ as the neighborhood of weight vector λ^i.

Step 1.3) Generate an initial population $\{x_1, x_2, \cdots, x_N\}$ randomly or using specific problem related methods within a given interval, calculate the corresponding target values of each individual in the population in the target space, and record them as FV_i, $FV_i = F(x_i)$, $i = 1, 2, \cdots, N$.

Step 1.4) Initialize $\mathrm{z} = (z_1, z_2, \cdots, z_m)^T$, where $z_i = min\{f_i(x_1), f_i(x_2), \cdots, f_i(x_N)\}$ $i = 1,2, \cdots, m$.

Step 1.5) Set the solution for each subproblem. For each λ_i , calculate the value of $g^{te}(x_k|\lambda_i, z)$, if $g^{te}(x_w|\lambda_i, z)$=$min\{g^{te}(x_k|\lambda_i, z)\}$, $i, k = 1,2, \cdots, N$, then x_w is the solution to the current subproblem i, denoted as $x_i = x_w$.

Step1.6) Initialize $\mathrm{z} = (z_1, z_2, \cdots, z_m)^T$, where $z_i = min\{f_i(x_1), f_i(x_2), \cdots, f_i(x_N)\}$, $i = 1,2, \cdots, m$.

Step 1.7) For any λ_i, calculate the value of $g^{te}(x_j|\lambda_i, z^*)$, taking the value of x_j when $g^{te}(x_j|\lambda_i, z^*)$ is the smallest as the solution to the subproblem i, $i, j = 1,2, \cdots, N$.

Step 2) The update process of individuals:

for i=1, 2, ⋯, N, do

Step 2.1) Set the current generation number is t, and use an adaptive strategy to select a offspring generation strategy to get y. This process is detailed in section 5.5.4.

Step 2.2) Modify the value of y. If necessary, use a problem-specific methods to repair y and noted as y'.

Step 2.3）Calculate the value of $R_i(x_i, y')$, the calculation method is shown in formulas (16) and (17). If the current value of $R_i(x_i, y') > 1$, and adjusting the mutation rate of the solution of subproblem i; then $n_i = n_i + 1$, and update its mutation rate; if s_i is an integer multiple of l, turn to Step 2.4 for Tabu search, otherwise turn to Step 2.5.

$$R_i(x, y) = \frac{g_i(y)}{g_{i-1}(x)} \tag{16}$$

$$g_i(x) = \sum_{j=1}^{m} \lambda_i^j \left| \frac{f_j(x) - z_j^*}{z_j^{nad} - z_j^*} \right| \tag{17}$$

Step 2.4) Set s_i=0, execute the Tabu search process. The best individual obtained from Tabu search is denoted as y'', if $g^{te}(y'|\lambda_i, z) \le g^{te}(x_j|\lambda_i, z)$, then $y' = y''$.

Step 2.5) Update the individuals corresponding to each weight vector in the neighborhood. For any $j \in B(i)$, compare the values of $g^{te}(y'|\lambda_i, z)$ and $g^{te}(x_j|\lambda_i, z)$, if $g^{te}(y'|\lambda_i, z) \le g^{te}(x_j|\lambda_i, z)$, then update $x_i = y'$.

Step 2.6) Update z. For any $j = 1, 2, \cdots, m$, if $f_j(y') < z_j$, update $z_j = f_j(y')$.

Step 2.7) Update the external archive set EP. If no individual dominates $f(y')$ in EP, add $f(y')$ to EP; If the number of individuals exceeds the size of EP, the k-nearest method(Coello et al., 2004) is used as the truncate strategy. If an individual in EP is dominated by $f(y')$, remove it from EP.

Step 3) Stopping conditions:

If the stopping conditions is met, the algorithm stops running and outputs EP. Otherwise, go to step 2.

The MOEA/D algorithm uses genetic operators to generate offspring individuals, which is suitable for optimization problems with multiple objectives and has good global search ability; The advantage of the tabu algorithm lies in neighborhood search, which achieves good local optimization ability by flexibly setting taboo lists and stopping criteria and has strong universality. Algorithm 1 combines Tabu search strategy and MOEA/D algorithm, which can consider both global and local optimization strategies, effectively improve the search ability of the algorithm, and reduce the convergence time of the algorithm. Meanwhile, the combination of the two can improve the search accuracy of the algorithm through parameter settings and optimization strategies. In summary, a hybrid algorithm combining MOEA/D with tabu algorithm can fully utilize the advantages of both methods, improve the search quality and efficiency of the algorithm, and have high application value in multi-objective optimization problems.

4.3 Population initialization and encoding

Many studies have confirmed that a good initial population has a promoting effect on the convergence speed and optimization performance of the algorithm(Rahnamayan et al., 2007). In order to improve the quality of the initial population, the SMOEA/D algorithm proposed in this paper uses a combination of NEH (Ham, 1983) and random generation methods to generate the initial population of studied problems, with each generation proportion accounting for 50%. As mentioned earlier, individuals encode using a segmented encoding, with the number of segments determined by the number of production lines, and the encoding length of each segment determined by the size of jobs on the corresponding production line. In the numerical experiment, the number of workpieces on each production line should be equal. Therefore, when randomly generating the initial individual encoding, the encoding segments representing the assembly sequence are first randomly generated, and then the encoding sequences of other segments are generated by perturbing assembly sequence. The population size is set based on the length of encoding.

The collaborative scheduling model for machining-assembly multiple production lines constructed in this paper involves the sorting of jobs to be processed on each production line and assembly line. The processing of jobs on each line is conducted independently and not affected by production activities on other production lines; Meanwhile, due to the kitting constraints of the assembling process on the assembly line, the sequence of subcomponents on each machining line will affect the output of the assembly production line by kitting constraints. Therefore, segmented encoding is used to represent individuals in the population, and each segment is encoded using an independent positive integer sequence. When there are L production lines, the number of jobs or product to be produced on machining line or assembly line l $(l = 1,2,\cdots,L)$ is n_l, the encoding form of the individual in population is$\{X_{11},X_{12},\cdots,X_{1n_1}|X_{21},X_{22},\cdots,X_{2n_2}|\cdots|X_{l1},X_{l2},\cdots,X_{ln_l}|\cdots|X_{L1},X_{L2},\cdots,X_{Ln_L}\}$, which is divided into L segments, representing the processing sequences of different jobs or product on L production lines, where the $L-th$ segment is the sequence on the assembly line.

To explain the segmented encoding method in detail, an example with 3 production lines and 5 jobs or products to be processed on each production line is given in Fig. 2. As shown in the figure, the encoding is divided into three parts, representing the three production lines in this multi-line collaborative scheduling problem. Part 1 is encoded as (3,1,5,4,2), corresponding to the processing sequence of jobs on the first production line; Part 2 is encoded as (1,5,3,2,4), corresponding to the processing sequence of jobs on the second production line; Part 3 is encoded as (1,2,4,5,3), corresponding to the assembly sequence of the products.

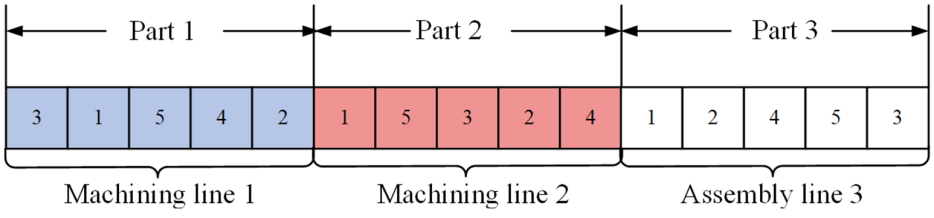

Fig. 2. The demo of encoding

4.4 The selection of aggregation functions

Traditional MOEA/D algorithms convert multi-objective optimization problems into a series of single-objective optimization subproblems using an aggregation functions, and their performance also depends on the aggregation function used(Zhang et al., 2010). In the SMOEA/D algorithm proposed in this paper, the Chebyshev method is used as the aggregation function, while a normalization method is used to eliminate the influence of dimensional error to obtain more uniform solution vectors (Li & Zhang, 2008). The Chebyshev method using normalization is shown in Formula (18).

$$min\ g^{te}(x|\lambda_i,z^*)=\max_{1\le j\le m}\left\{(\lambda_i^j\left|\frac{f_j(x)-z_j^*}{Z_j^{nad}-z_j^*}\right|\right\} \tag{18}$$

where$z^{nad}$is the nadir point in the object space,$z^{nad}=(z_1^{nad},z_2^{nad},\cdots,z_m^{nad})^T$,$z_j^{nad}=max\{f_j(x)|x\in\varphi\}$, φ is the set of all

the Pareto optimal solutions for the problem.

4.5 The updating process of population individuals

The search process of metaheuristic algorithms has randomness, and the search strategies in metaheuristic algorithms may different. However, corresponding mechanisms are designed to guide individuals to jump out of local optima to obtain global optimization results. In the proposed SMOEA/D algorithm, adaptive changes in the probability of parent selection strategies being selected are used to guide the search direction as much as possible towards the better region, to accelerate the convergence speed of the algorithm. In the early stage of algorithm operation, increasing the probability of random search is beneficial for expanding the search range，and avoiding falling into local optima; In the later stage, the population quality has converged to a certain extent, and selecting better individuals as parents to generate offspring can enhance the convergence of the algorithm and accelerate the convergence speed. Therefore, the application of adaptive parent selection strategy in the SMOEA/D algorithm has improved the optimization ability of the algorithm.

In general, if the solution's quality of subproblem has not improved after multiple searches, there are two situations: the individual falls into a local optimum, or the subproblem has already obtained the optimal solution. Here, the following strategies are adopted to deal with these two situations: increase the mutation rate. If it has not been improved, Tabu search is performed on the neighborhood of the individual to determine whether there is still room for improvement. The search process is shown in Fig. 3.

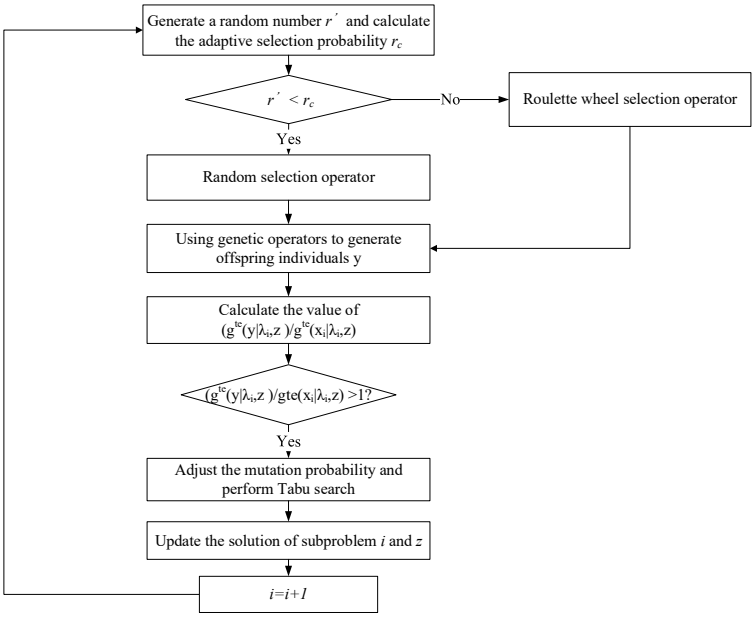

Fig. 3. The update process of individuals in population

Next, we will provide a detailed introduction to the steps and process of updating individuals in the population in the SMOEA/D algorithm.

4.5.1 Adaptive Parent Individual Selection Strategy

The SMOEA/D algorithm employs two different parent selection strategies: a random selection strategy and a roulette wheel selection strategy. The former selects indexes p and q randomly from the neighborhood B(i) of sub-problem i, and generates offspring individuals y_1 and y_2 using the genetic operator based on x_p and x_q, Take the individual with smaller g^{te} value as y. When using the roulette wheel selection strategy, the fitness value of individual x in the neighborhood of sub-problem i is calculated using formulas (19) to (21). For sub-problem i, the probability that individual x_i' in $B(i)$ is selected as a parent is shown in formula (21), where $x_i' \in B(i)$.

$$v^{te}(\lambda_i, x) = \sum_{j=1}^{m} (\lambda_i^j \left| \frac{f_j(x) - z_j^*}{Z_j^{nad} - z_j^*} \right|) \tag{19}$$

$$Fit(\lambda_i, x) = \frac{1}{v^{te}(\lambda_i, x)} \tag{20}$$

$$P\left(\lambda_i, x_i^{'}\right) = \frac{Fit\left(\lambda_i, x_i^{'}\right)}{\sum_{k=i_1}^{iT} Fit(\lambda_i, x_k)} \tag{21}$$

During the iteration process of the SMOEA/D algorithm, randomly generate a numerical value r' in [0,1], and then compared with the adaptive selection probability r_c. If $r' < r_c$, the random selection method is used to select the parent individual; Otherwise, calculate the fitness of each individual, and then select the parent individuals by roulette wheel selection strategy in term of the fitness value. The calculation method of the adaptive selection probability r_c is shown in formula (22).

$$r_c = 1 - \frac{gen}{gen\_max} \tag{22}$$

where gen is the current generations, and gen_{max} the maximum iteration times. As the algorithm runs, the value of r_c dynamically increases, and adapts to select the parent selection strategy to generate offspring individuals according to the value of r_c.

4.5.2 Generation of offspring individuals

When the parent individuals are selected, genetic operators are used to each segment in the encoding to generate offspring. In the numerical experiment of the paper, the Partial-mapped crossover (PMX) operator is used to generate offspring individuals. PMX is commonly used as a crossover operator in generating offspring for various algorithms that represent individuals in a population as integer sequences. It can ensure that each numerical value in the generated offspring individual only appears once, there will be no duplicate numerical values in the offspring sequence by this crossover strategy. Therefore, PMX is commonly used to production scheduling problems. The basic steps of PMX operator are as follows.

Step 1: Randomly generate two indexes, and the sub-sequence between the two index points is the part to be partially crossed.
Step 2: Swap the positions of the two sets of natural number sequences while keeping the values of the other positions unchanged.
Step 3: Check for duplicate values and perform replacements. Traverse the encoding values of the non-swapped parts in an individual, and then search for duplicate values in the replacement part. If so, find their corresponding replaced values. Repeat this step until there are no duplicate values.

Taking individuals {1,2,3,4,5,6,7,8,9} and {5,4,6,9,2,1,7,8,3} as parents, the process of generating offspring individuals using the PMX operator is shown in Fig. 4. After the generation of offspring individuals, determine whether to screen or modify them based on the characteristics of the problem; Then calculate the g^{te} of each offspring individual, and select the individual with the lowest g^{te} value as y.

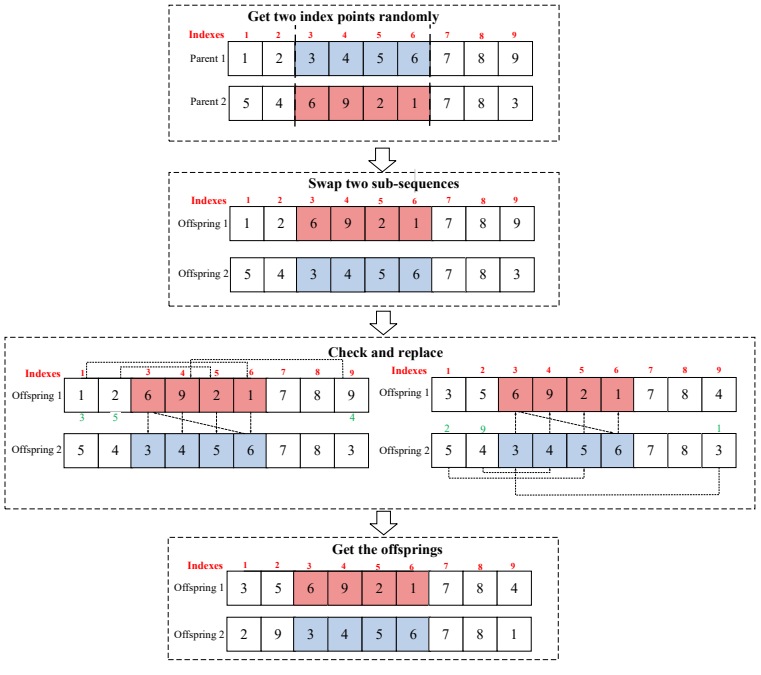

Fig. 4. PMX crossover operator

After obtaining new solution y, using formula (16) to calculate the value of $R_i(x_i, y)$. If the solution of the current subproblem

is not improved, use formula (23) to adjust its mutation rate of subproblem i.

$$P_{c,i}^{'} = P_c + (1 - P_c)\frac{|v^{te}(\lambda_i, x_i) - v^{te}(\lambda_i, y)|}{v^{te}(\lambda_i, x_i)} \tag{23}$$

Next, determine whether to perform a local search process. If the subproblem i is not improved in the generations of n_i, and n_i is an integer multiple of l, then the tabu search process is executed, and the execution steps of this process are shown in algorithm 2.

Algorithm 2 :The update process of the solution for the subproblem

Step 1) Set the current generation number is t, the current individual $x = y'$, the global optimal solution $x_i = y'$; Initialize tabu memory $H(x) = \phi$, the current generation number is labeled as k=1, and the maximum generation number is I_{max}.

Step 2) If the Tabu search stopping condition is met, stop the tabu search process and turn to the optimization of the subproblem $(i + 1)$ or the $(k + 1) - th$ iteration of SMOEA/D; Otherwise, proceed to step 3.

Step 3) Use swap as move Operator to generate the neighborhood $N(x, k)$ of the current individual x, and select an appropriate number of candidate solutions based on the scale of the problem, represented by $N'(x, k)$.

Step 4) Calculate the g^{te} of each individual in the candidate solution set $N'(x, k)$, and define $x' = \{x''|g^{te}(x''|\lambda_i, z) = min\{g^{te}(x_j|\lambda_i, z)\}, x_j \epsilon N'(x, k)\}$, if x' meets the aspiration criteria, update $x = x'$, $x_i = x'$, and update the aspiration criteria, then turn to Step 6; Otherwise, turn to Step 5.

Step 5) Check the tabu attributes of each individual in $N'(x, k)$, define x'=$\{x''|g^{te}(x''|\lambda_i, z) = min\{g^{te}(x_j|\lambda_i, z)\}, x_j \epsilon N'(x, k)/H\}$, and update tabu memory $H(x, k)$;

Step 7) Turn to Step 2.

5. Experimental design and Result Analysis

This paper investigates the collaborative scheduling problem of machining-assembly multi production lines, constructs a collaborative scheduling model for machining-assembly multi production lines, and proposes the SMOEA/D algorithm to solve the problem. In order to verify the effectiveness of the proposed SMOEA/D algorithm in solving such cooperative scheduling problems, SMOEA/D is compared with the result of three classical multi-objective optimization algorithms NSGAII (Deb et al., 2002), MOEA/D (Li & Zhang, 2008), and MOPSO (Coello et al., 2004) on each instance in numerical experiment. All algorithms are implemented using the Java language on the MOEA Framework platform. The experimental testing environment is Apple M1 pro, 16GB of memory, and macOS Ventura 13.0.1 operating system.

5.1 Instances generation

To test the effectiveness of the proposed improved algorithm in solving the collaborative scheduling problem of machining-assembly, two sets of numerical experiments were designed in this section, corresponding to two situations: the equal number of operations on each production line and the unequal number of processes on each production line. For the convenience of later description, they were respectively used as Experiment I and Experiment II. In each group of experiments, instances of different scales are generated through the changes of three factors: the number of jobs to be processed on each production line, the number of machining lines, and the number of operations in each machining line.

In Experiment I, the jobs for each production line were set at three different scales: 10, 20, and 50. The number of production lines was set at 2, 5, and 10, respectively. The number of operations for each production line was considered at 4, 8, and 12. Therefore, a total of 27 instances were generated. In Experiment II, we mainly considered the impact of the differences in the number of operations in each production line on the experimental results. Three different degrees of differentiation were set, as shown in Table 1. The other two factors had the same numerical settings as in Experiment I. In the two groups of experiments, each instance was marked in the form of "$l - m - n$", where "l" represents the quantity of production line (including assembly line), and "n" represents the number of jobs on each production line; The meaning of "m" in Experiment I is different from that in Experiment II: In Experiment I, "m" represents the number of operations in each production line; In Experiment II, "m" represents the degree of differentiation in the number of processes in different production lines, "S" represents a relatively close number of operations in each production line, "L" represents a significant difference in the number of operations, and "M" represents the difference in the number of operations in each production line between the first two. When defining a instance set, the set of operation quantities of each production line is obtained from Table 1 according to the values of "l" and "m". The processing (assembly) times of jobs or products of different processes on each production line (including machining and assembly lines) are random numbers taken from a uniform distribution in the interval [17, 46].

Table 1
Setting of the operations for production lines in different instances

Code	Range	The number of operations on each production line
S	DU[9,12]	12,9
		10,12,9,9,10
		10,10,12,9,10,11,10,10,12,12
M	DU[6,12]	11,6
		12,6,6,10,7
		7,8,12,9,6,12,11,10,9,12
L	DU[3,12]	3,12
		9,3,6,7,12
		12,11,8,7,8,7,3,9,3,8

5.2 Parameter settings

In general, different parameter settings can affect the optimization performance of the algorithm (Hao et al., 2013). In order to test the performance of the proposed algorithm in solving the cooperative scheduling problem, a set of instances were generated by changing the number of production lines, jobs and operations in each production line, and a large number of comparative experiments were carried out. For the algorithms used in these comparative experiments, most parameters can be set to fixed values, but some parameters, such as population size and evaluation times, are more sensitive to the size of the instances, and must be set reasonably according to the size of the current instance to better present the optimization performance of the algorithm. In these two groups of experiments, the scale of the production line is {2,5,10}, and the scale of the jobs on each production line is {10,20,50}. Therefore, we can see that the job in the instances has seven different scales: 20, 40, 50, 100, 200, 250, and 500. So, the corresponding population size is set to 50, 80, 100, 150, 300, 350, and 650 in the four algorithms, and the maximum number of evaluation times is 500, 500, 1000, 1000, 1200, 2000, and 2000, respectively. In the SMOEA/D and MOEA/D algorithms, corresponding neighborhood sizes are 0.1N, where N is the population size. The settings of other parameters for the four algorithms are shown in Table 2, where N represents the population size and L represents the number of production lines.

Table 2
Parameter settings for four multi-objective algorithms

Parameters	Values
Crossover rate:	P_c =0.9 (For SMOEA/D, NSGAII and MOEA/D)
Mutation rate:	P_m =0.1 (For SMOEA/D, NSGAII and MOEA/D)
External population:	N (For SMOEA/D, NSGAII, MOEA/D and MOPSO)
Neighborhood size$(N(x))$:	N (For SMOEA/D and MOEA/D)
The number of tabu memories:	L (For SMOEA/D)
Tabu length:	$\left\lceil \sqrt{\frac{N}{L}} \right\rceil$(For SMOEA/D)
Generations of tabu search:	50 (For SMOEA/D)
Inertia weight:	0.4 (For MOPSO)
Acceleration coefficients:	$c_1 = c_2 = 0.2$ (For MOPSO)

In two groups of experiments, SMOEA/D algorithm and three comparison methods were used to independently run 30 times on each instance.

5.3 Analysis of Experiment I Results

The statistical data of the running results of each algorithm are shown in Table 3. Table 3 shows the statistical values of SMOEA/D algorithm, NSGA-II (Deb et al., 2002), MOEA/D (Li & Zhang, 2008) and MOPSO (Coello et al., 2004) on evaluation indicators GD (Zitzler et al., 2003), IGD(S. W. Jiang et al., 2014) and Spread (Deb et al., 2002) after 30 independent runs on each instance. Each indicator value for instance gives the average and standard deviation of the statistical values obtained on 30 independent runs. From Table 3, the statistical results obtained by the SMOEA/D algorithm are superior to the other three algorithms in most instances. Only in “10-4-10” and “5-4-50” does NSGA-II perform better than SMOEA/D in terms of the convergence indicator GD. Especially in the comprehensive evaluation indicator IGD, the SMOEA/D algorithm is superior to the other three algorithms in all 27 instances. The SMOEA/D algorithm transforms the multi-objective optimization problem into multiple single-objective optimization sub-problems, guiding the optimization process of each sub-problem through weight vectors. This processing strategy can maintain the diversity and uniformity of the population, enhancing the global search ability of the algorithm. Moreover, the adaptive crossover probability generation method can balance local optimization and global development capabilities, enhancing the solution performance of the algorithm. In summary, the proposed SMOEA/D algorithm has superiority in solving multi-line collaborative scheduling problems.

Table 3
The average and standard deviation of indicators obtained from SMOEA/D, NSGA-II, MOEA/D, and MOPSO algorithms (with the same number of operations in each production line)

Instances	SMOEA/D			NSGA-II			MOEA/D			MOPSO		
	GD	IGD	Δ	GD	IGD	Δ	GD	IGD	Δ	GD	IGD	Δ
2-4-10	**9.80E-2**	**1.33E-1**	**7.37E-1**	1.28E-1	2.26E-1	7.79E-1	1.53E-1	2.77E-1	7.59E-1	1.62E-1	3.24E-1	8.02E-1
	1.54E-2	**1.71E-2**	7.54E-2	1.76E-2	3.19E-2	9.93E-2	2.25E-2	3.65E-2	7.98E-2	2.15E-2	3.72E-2	**7.34E-2**
2-4-20	**1.01E-1**	**1.16E-1**	**7.37E-1**	1.19E-1	1.30E-1	8.12E-1	1.23E-1	1.58E-1	8.43E-1	1.27E-1	1.91E-1	7.59E-1
	3.76E-2	3.70E-2	**2.70E-2**	4.68E-2	**3.61E-2**	5.35E-2	5.63E-2	4.29E-2	4.87E-2	4.69E-2	4.59E-2	5.12E-2
2-4-50	**2.91E-2**	**2.87E-2**	**8.26E-1**	3.19E-2	4.58E-2	8.80E-1	3.07E-2	4.81E-2	8.46E-1	3.40E-2	5.05E-2	8.80E-1
	1.83E-3	**6.58E-3**	**5.15E-2**	8.31E-3	1.18E-2	7.10E-2	8.45E-3	1.26E-2	8.27E-2	8.02E-3	1.35E-2	1.09E-1
2-8-10	**2.41E-1**	**2.27E-1**	**6.93E-1**	2.68E-1	3.88E-1	7.63E-1	3.02E-1	4.65E-1	7.96E-1	2.86E-1	5.25E-1	8.03E-1
	4.05E-2	**7.44E-2**	7.02E-2	5.37E-2	1.05E-1	6.82E-2	6.54E-2	1.45E-1	7.80E-2	6.30E-2	1.55E-1	**4.13E-2**
2-8-20	**3.66E-2**	**3.34E-2**	**7.50E-1**	4.13E-2	4.19E-2	8.10E-1	5.00E-2	6.08E-2	8.50E-1	5.01E-2	6.87E-2	8.10E-1
	6.90E-3	**3.71E-3**	**1.17E-2**	1.45E-2	4.48E-3	8.07E-2	9.75E-3	7.46E-3	6.95E-2	1.34E-2	8.49E-3	7.29E-2
2-8-50	**4.99E-2**	**6.14E-2**	**8.37E-1**	6.07E-2	6.82E-2	8.84E-1	6.54E-2	7.90E-2	8.75E-1	6.65E-2	8.75E-2	8.90E-1
	9.75E-3	**1.66E-2**	**4.76E-2**	1.58E-2	2.00E-2	5.85E-2	2.09E-2	2.16E-2	9.01E-2	1.81E-2	2.40E-2	5.60E-2
2-12-10	**3.60E-1**	**2.11E-1**	**7.11E-1**	4.02E-1	2.42E-1	8.17E-1	4.22E-1	3.44E-1	8.56E-1	4.27E-1	3.64E-1	9.15E-1
	1.16E-2	**5.96E-2**	**5.01E-2**	8.97E-2	6.88E-2	8.17E-2	1.28E-1	8.69E-2	6.14E-2	8.25E-2	8.88E-2	8.07E-2
2_12_20	**2.47E-2**	**3.23E-2**	**7.16E-1**	2.53E-2	3.34E-2	7.32E-1	3.34E-2	4.42E-2	7.19E-1	3.65E-2	5.03E-2	8.01E-1
	6.48E-3	**2.76E-3**	**6.58E-2**	**2.80E-3**	2.82E-3	6.77E-2	7.54E-3	3.62E-3	8.54E-2	5.21E-3	4.04E-3	7.94E-2
2-12-50	**3.16E-2**	**2.86E-2**	**7.55E-1**	3.34E-2	3.25E-2	7.79E-1	3.38E-2	4.64E-2	8.14E-1	3.43E-2	4.95E-2	8.23E-1
	1.44E-2	**1.24E-2**	5.86E-2	1.71E-2	1.50E-2	**4.70E-2**	1.78E-2	2.31E-2	5.12E-2	2.05E-2	2.44E-2	7.11E-2
5-4-10	**5.29E-1**	**7.15E-1**	**8.00E-1**	5.94E-1	9.40E-1	8.70E-1	6.78E-1	1.50E+0	8.33E-1	7.08E-1	1.57E+0	8.39E-1
	7.74E-2	**1.26E-1**	**4.39E-2**	1.19E-1	1.40E-1	5.37E-2	1.32E-1	2.19E-1	4.70E-2	1.35E-1	2.41E-1	5.82E-2
5-4-20	**5.70E-2**	**4.29E-2**	**7.29E-1**	6.15E-2	7.99E-2	7.75E-1	6.53E-2	8.44E-2	8.33E-1	6.65E-2	8.86E-2	8.48E-1
	2.62E-2	**1.44E-2**	**2.51E-2**	4.89E-2	3.76E-2	1.01E-1	5.82E-2	3.98E-2	4.04E-2	5.84E-2	4.01E-2	4.78E-2
5-4-50	1.67E-2	**2.26E-2**	**7.24E-1**	**1.54E-2**	2.46E-2	7.73E-1	1.81E-2	2.92E-2	7.50E-1	1.92E-2	3.27E-2	7.82E-1
	3.58E-3	**1.81E-3**	**3.14E-2**	**3.16E-3**	4.10E-3	6.34E-2	1.85E-3	4.71E-3	4.15E-2	2.47E-3	4.63E-3	4.09E-2
5-8-10	**3.32E-1**	**1.43E-1**	**7.19E-1**	3.70E-1	2.39E-1	8.30E-1	3.79E-1	4.01E-1	9.06E-1	3.95E-1	4.25E-1	9.28E-1
	1.53E-1	**2.05E-2**	**3.08E-2**	1.60E-1	6.61E-2	5.49E-2	1.53E-1	9.06E-2	8.88E-2	1.61E-1	1.01E-1	1.03E-1
5-8-20	**4.24E-2**	**5.30E-2**	**6.95E-1**	4.44E-2	6.49E-2	7.59E-1	5.04E-2	7.40E-2	8.23E-1	5.00E-2	7.75E-2	7.77E-1
	5.52E-3	**9.81E-3**	7.11E-2	6.72E-3	1.27E-2	7.94E-2	7.50E-3	1.15E-2	5.86E-2	9.66E-3	1.25E-2	**5.72E-2**
5-8-50	**2.08E-2**	**2.27E-2**	**7.35E-1**	3.36E-2	4.14E-2	7.97E-1	3.46E-2	4.36E-2	8.06E-1	3.53E-2	4.60E-2	8.32E-1
	5.83E-3	**7.97E-3**	**1.20E-2**	2.09E-2	1.59E-2	6.53E-2	1.98E-2	1.68E-2	6.86E-2	2.20E-2	1.75E-2	4.47E-2
5-12-10	**1.04E-1**	**1.38E-1**	**7.50E-1**	2.35E-1	1.70E-1	8.63E-1	1.61E-1	2.29E-1	7.70E-1	1.93E-1	3.23E-1	8.23E-1
	4.46E-2	**4.00E-2**	**5.37E-2**	6.10E-2	6.57E-2	1.59E-1	6.02E-2	8.60E-2	1.08E-1	7.73E-2	1.22E-1	6.95E-2
5-12-20	**3.53E-2**	**3.66E-2**	**7.92E-1**	3.89E-2	3.69E-2	7.96E-1	3.83E-2	5.41E-2	8.08E-1	4.04E-2	5.79E-2	8.03E-1
	1.02E-2	**1.35E-2**	**2.27E-2**	1.54E-2	5.20E-2	1.12E-1	1.14E-2	1.65E-2	7.26E-2	1.10E-2	1.70E-2	4.03E-2
5-12-50	**3.08E-2**	**3.09E-2**	**7.44E-1**	3.52E-2	3.65E-2	7.74E-1	3.77E-2	4.27E-2	8.00E-1	3.76E-2	4.90E-2	8.45E-1
	7.44E-3	**6.57E-3**	**4.49E-2**	1.17E-2	7.68E-3	5.19E-2	1.37E-2	1.04E-2	5.15E-2	1.39E-2	1.35E-2	7.16E-2
10-4-10	3.02E-1	**1.74E-1**	7.17E-1	**2.63E-1**	2.16E-1	**7.02E-1**	3.23E-1	2.89E-1	8.03E-1	3.87E-1	3.63E-1	7.90E-1
	2.84E-1	**4.29E-2**	**7.32E-2**	**1.96E-1**	5.62E-2	1.37E-1	1.61E-1	7.10E-2	1.31E-1	9.99E-2	7.62E-2	7.40E-2
10-4-20	**3.52E-2**	**8.25E-2**	**7.22E-1**	3.68E-2	8.98E-2	7.51E-1	3.66E-2	9.48E-2	7.60E-1	3.82E-2	1.01E-1	7.71E-1
	1.04E-3	**6.05E-3**	**4.03E-2**	6.53E-3	1.34E-2	4.07E-2	7.81E-3	1.40E-2	6.23E-2	9.13E-3	1.96E-2	6.14E-2
10_4_50	**1.57E-2**	**2.63E-2**	**7.14E-1**	1.82E-2	2.87E-2	7.67E-1	1.88E-2	3.28E-2	7.55E-1	1.89E-2	3.46E-2	7.85E-1
	2.75E-3	**3.93E-3**	**3.24E-2**	3.91E-3	4.62E-3	4.47E-2	2.80E-3	6.98E-3	5.91E-2	2.84E-3	6.70E-3	6.72E-2
10-8-10	**3.70E-1**	**3.28E-1**	**7.69E-1**	4.24E-1	3.85E-1	8.14E-1	3.91E-1	5.73E-1	8.11E-1	4.50E-1	6.77E-1	9.17E-1
	1.16E-2	**1.63E-1**	**7.47E-2**	1.08E-1	2.06E-1	8.50E-2	9.97E-2	2.13E-1	8.75E-2	5.46E-2	2.20E-1	9.19E-2
10-8-20	**7.91E-2**	**6.50E-2**	**7.39E-1**	9.15E-2	8.50E-2	7.81E-1	9.21E-2	1.08E-1	7.88E-1	9.71E-2	1.37E-1	7.73E-1
	2.05E-2	**1.01E-2**	**7.17E-3**	2.07E-2	1.56E-2	4.19E-2	3.65E-2	2.05E-2	4.41E-2	2.88E-2	2.89E-2	5.69E-2
10-8-50	**2.26E-2**	**2.98E-2**	**6.04E-1**	2.70E-2	3.61E-2	6.99E-1	3.42E-2	4.23E-2	7.92E-1	3.20E-2	4.99E-2	6.51E-1
	2.16E-3	**7.74E-3**	**3.44E-2**	7.61E-3	9.51E-3	6.39E-2	1.46E-2	1.34E-2	4.58E-2	1.32E-2	1.46E-2	7.07E-2
10-12-10	**6.05E-1**	**6.36E-1**	**7.85E-1**	6.42E-1	7.65E-1	7.82E-1	6.94E-1	1.12E+0	8.64E-1	6.83E-1	1.21E+0	9.56E-1
	1.78E-1	**3.42E-1**	**1.03E-2**	2.43E-1	4.54E-1	1.12E-1	2.80E-1	6.18E-1	7.37E-2	2.47E-1	6.64E-1	8.15E-2
10-12-20	**6.24E-2**	**6.15E-2**	**7.25E-1**	6.63E-2	6.66E-2	7.70E-1	7.85E-2	8.16E-2	7.70E-1	8.25E-2	9.56E-2	7.71E-1
	2.47E-2	**2.55E-2**	**3.97E-2**	5.74E-2	2.91E-2	5.19E-2	3.82E-2	3.61E-2	5.86E-2	3.52E-2	4.27E-2	6.53E-2
10-12-50	**2.24E-2**	**3.36E-2**	**6.37E-1**	2.67E-2	3.76E-2	7.16E-1	2.93E-2	4.96E-2	8.47E-1	3.14E-2	5.23E-2	8.35E-1
	3.76E-3	**7.69E-3**	**1.53E-2**	5.10E-3	8.47E-3	3.76E-2	3.03E-3	1.09E-2	7.29E-2	**2.73E-3**	1.13E-2	4.56E-2
Statistical results	**25/23**	**27/26**	**26/23**	**2/3**	**0/1**	**1/1**	**0/0**	**0/0**	**0/0**	**0/1**	**0/0**	**0/3**

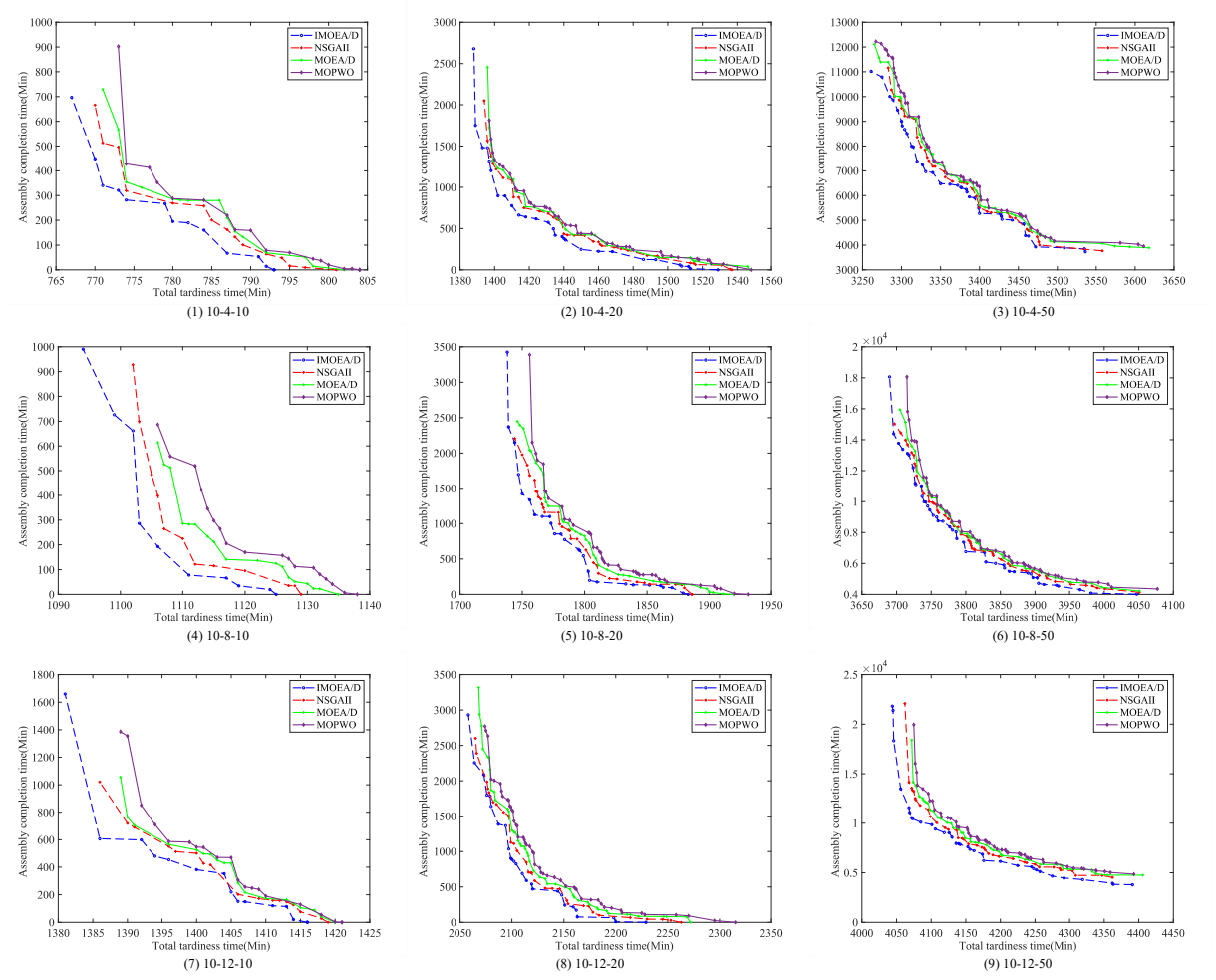

Fig. 5. Convergence curves of partial instances with the best IGD run by different MOEAs in experiment I

To observe the experimental results more intuitively and analyze the optimization performance of these algorithms, convergence curves of Pareto solutions corresponding to the optimal IGD obtained by four different algorithms on each instance were generated, some of the convergence curves are shown in Fig. 5. Clearly, compared with other MOEAs, the SMOEA/D algorithm proposed in this chapter can obtain solutions with better convergence and diversity in most test problems. When the number of jobs is small, there is a big difference in the convergence curve among the four algorithms. As the scale of the job's increases, the convergence curves of the four algorithms converge, but in most test problems, the evaluation indicators are better than the three reference algorithms.

5.4 Analysis of Experiment II Results

In Experiment II, the numbers of production processes in each production line for each instance were randomly generated, while the other data was the same as in Experiment I. Table 4 records the GD, IGD, and Spread values obtained from 30 independent runs of four algorithms (SMOEA/D, NSGA-II, MOEA/D, and MOPSO) on 27 instances where the numbers of production operations in each production line were not equal. Each indicator value was calculated for both its mean and standard deviation, and the best indicator values are shown in bold.

Table 4
The average and standard deviation of indicators obtained from SMOEA/D, NSGA-II, MOEA/D, and MOPSO algorithms (when the number of operations on each production line is not equal)

Instances	SMOEA/D			NSGAII			MOEA/D			MOPSO		
	GD	IGD	Δ	GD	IGD	Δ	GD	IGD	Δ	GD	IGD	Δ
2-S-10	**1.17E-1**	**1.39E-1**	**7.36E-1**	1.45E-1	1.90E-1	7.84E-1	1.48E-1	2.03E-1	7.78E-1	1.62E-1	2.49E-1	7.84E-1
	7.93E-2	**9.42E-2**	**6.73E-2**	8.47E-2	1.32E-1	8.32E-2	8.40E-2	1.38E-1	9.15E-2	8.37E-2	1.66E-1	7.06E-2
2-S-20	**2.40E-2**	**2.30E-2**	**5.07E-1**	3.55E-2	2.76E-2	7.55E-1	4.27E-2	3.26E-2	7.71E-1	4.52E-2	3.64E-2	7.64E-1
	9.14E-3	**1.19E-2**	**4.29E-2**	1.13E-2	1.21E-2	1.09E-1	1.18E-2	1.33E-2	5.93E-2	1.48E-2	1.52E-2	8.89E-2
2-S-50	**1.84E-2**	**1.11E-2**	**7.81E-1**	2.27E-2	2.34E-2	8.65E-1	2.22E-2	1.61E-2	8.09E-1	2.17E-2	1.98E-2	8.48E-1
	4.96E-3	**2.25E-3**	**3.65E-2**	6.37E-3	6.13E-3	3.84E-2	5.46E-3	2.59E-3	1.27E-1	5.50E-3	4.72E-3	1.15E-1
2-M-10	1.74E-1	**1.78E-1**	**7.17E-1**	1.97E-1	1.90E-1	8.54E-1	**1.66E-1**	2.35E-1	7.72E-1	2.09E-1	3.55E-1	7.67E-1
	4.32E-2	**7.01E-2**	**1.38E-1**	9.65E-2	7.65E-2	1.86E-1	5.13E-2	8.48E-2	4.38E-1	5.71E-2	1.42E-1	6.85E-1
2-M-20	**1.66E-2**	**1.44E-2**	7.54E-1	1.74E-2	1.70E-2	**7.17E-1**	2.40E-2	1.87E-2	7.44E-1	2.39E-2	2.39E-2	7.50E-1
	2.78E-3	**1.90E-3**	**1.26E-2**	4.33E-3	2.17E-3	4.41E-2	6.76E-3	1.94E-3	3.70E-2	3.41E-3	2.17E-3	6.62E-2
2-M-50	**1.03E-2**	**1.05E-2**	**6.98E-1**	1.55E-2	1.26E-2	7.87E-1	1.57E-2	1.50E-2	7.98E-1	1.68E-2	1.74E-2	7.63E-1
	5.75E-3	**1.85E-3**	**5.37E-2**	7.82E-3	1.89E-3	5.68E-2	5.27E-3	2.16E-3	6.95E-2	9.30E-3	2.00E-3	5.58E-2
2-L-10	1.44E-1	**9.06E-2**	**6.82E-1**	**1.39E-1**	1.25E-1	7.33E-1	1.80E-1	1.63E-1	8.10E-1	2.01E-1	2.13E-1	7.71E-1
	5.74E-2	**3.43E-2**	**6.71E-2**	6.83E-2	5.19E-2	7.51E-2	6.82E-2	5.46E-2	1.04E-1	6.66E-2	1.10E-1	9.52E-2
2-L-20	**1.60E-2**	**1.33E-2**	**3.78E-1**	2.40E-2	1.76E-2	7.77E-1	2.15E-2	2.13E-2	7.22E-1	2.20E-2	2.60E-2	5.72E-1
	2.52E-3	**2.69E-3**	**6.50E-2**	1.11E-2	4.49E-3	8.70E-2	4.11E-3	5.38E-3	7.36E-2	6.91E-3	3.31E-3	9.91E-2
2-L-50	**1.33E-2**	**1.30E-2**	**7.27E-1**	1.59E-2	1.54E-2	7.58E-1	1.52E-2	1.69E-2	7.32E-1	2.12E-2	2.08E-2	7.95E-1
	1.01E-2	**3.88E-3**	**2.70E-2**	1.37E-2	7.70E-3	8.43E-2	6.76E-3	6.91E-3	6.38E-2	1.10E-2	9.20E-3	3.79E-2
5-S-10	**3.26E-2**	**6.59E-2**	**4.45E-1**	4.70E-2	2.82E-1	4.70E-1	5.97E-2	9.00E-2	5.57E-1	7.47E-2	1.11E-1	6.12E-1
	1.07E-2	**1.58E-2**	**5.96E-2**	1.51E-2	8.70E-2	1.04E-1	1.56E-2	1.72E-2	8.57E-2	2.05E-2	2.50E-2	1.35E-1
5-S-20	**3.45E-2**	**2.05E-2**	**7.29E-1**	4.11E-2	2.64E-2	7.72E-1	4.26E-2	4.67E-2	7.81E-1	4.28E-2	4.96E-2	7.93E-1
	1.68E-3	**1.77E-3**	**1.29E-2**	1.77E-3	2.99E-3	9.40E-2	9.15E-3	7.56E-3	4.27E-2	9.14E-3	7.07E-3	4.28E-2
5-S-50	**1.30E-2**	**9.50E-3**	**7.31E-1**	1.59E-2	1.05E-2	7.42E-1	1.72E-2	1.18E-2	7.81E-1	1.67E-2	1.49E-2	7.53E-1
	6.25E-3	**2.33E-3**	**6.88E-2**	7.53E-3	2.44E-3	7.49E-2	8.01E-3	3.46E-3	1.03E-1	6.39E-3	3.50E-3	9.67E-2
5-M-10	**5.37E-2**	**1.29E-1**	7.76E-1	7.64E-2	1.36E-1	7.14E-1	8.96E-2	1.51E-1	7.33E-1	6.30E-2	1.34E-1	**5.33E-1**
	1.42E-2	**1.40E-2**	**5.14E-2**	3.12E-2	3.08E-2	7.79E-2	4.27E-2	2.53E-2	5.21E-2	2.61E-2	1.72E-2	6.98E-2
5-M-20	**1.48E-2**	2.33E-2	**4.78E-1**	3.44E-2	2.13E-2	7.70E-1	3.02E-2	2.34E-2	7.76E-1	4.01E-2	**2.21E-2**	7.37E-1
	4.57E-3	**2.26E-3**	**3.95E-2**	1.63E-2	7.49E-3	1.48E-1	1.20E-2	6.07E-3	5.69E-2	2.44E-2	2.78E-3	1.02E-1
5-M-50	**1.46E-2**	**2.12E-2**	**7.01E-1**	1.94E-2	2.25E-2	8.01E-1	1.59E-2	2.29E-2	7.56E-1	2.17E-2	2.45E-2	7.85E-1
	1.01E-2	**3.98E-3**	**5.08E-2**	**9.49E-3**	7.63E-3	9.23E-2	4.70E-3	4.28E-3	6.25E-2	8.71E-3	4.84E-3	9.63E-2
5-L-10	**1.09E-1**	**1.52E-1**	**7.41E-1**	1.15E-1	**1.39E-1**	7.52E-1	1.37E-1	1.70E-1	7.87E-1	1.48E-1	1.82E-1	7.72E-1
	6.57E-2	**3.21E-2**	**4.84E-2**	7.17E-2	6.30E-2	5.83E-2	1.00E-1	7.75E-2	5.97E-2	1.10E-1	8.41E-2	7.93E-2
5-L-20	**1.45E-2**	**1.77E-2**	**6.71E-1**	2.26E-2	2.81E-2	7.24E-1	2.24E-2	2.42E-2	6.69E-1	2.55E-2	2.90E-2	7.05E-1
	3.27E-3	**1.25E-2**	**4.85E-2**	5.13E-3	2.82E-3	6.45E-2	3.69E-3	2.19E-3	4.26E-2	4.38E-3	4.26E-3	5.35E-2
5-L-50	**1.65E-2**	1.79E-2	**5.09E-1**	1.80E-2	2.04E-2	7.89E-1	1.86E-2	**1.76E-2**	7.76E-1	1.92E-2	2.78E-2	8.34E-1
	5.90E-3	**2.55E-3**	**3.40E-2**	8.44E-3	2.84E-3	7.25E-2	8.32E-3	3.46E-3	3.93E-2	7.93E-3	3.99E-3	4.42E-2
10-S-10	**1.36E-1**	**1.66E-1**	**4.53E-1**	2.64E-1	2.40E-1	6.90E-1	2.44E-1	3.10E-1	8.38E-1	1.39E-1	1.10E+00	4.92E-1
	6.66E-2	**6.49E-2**	1.07E-1	1.71E-1	1.01E-1	1.77E-1	1.23E-1	1.25E-1	**8.06E-2**	5.00E-2	5.08E-1	1.08E-1
10-S-20	**1.96E-2**	**1.53E-2**	**6.97E-1**	2.28E-2	2.05E-2	7.23E-1	**1.95E-2**	2.33E-2	7.17E-1	3.09E-2	3.03E-2	7.40E-1
	4.08E-3	**2.77E-3**	**5.75E-2**	6.57E-3	2.81E-3	6.35E-2	4.41E-3	6.04E-3	6.01E-2	8.81E-3	7.36E-3	7.59E-2
10-S-50	**1.38E-2**	**1.18E-2**	**7.03E-1**	1.63E-2	1.21E-2	7.14E-1	2.09E-2	1.40E-2	8.28E-1	1.96E-2	2.14E-2	8.68E-1
	4.05E-3	2.23E-3	**6.16E-2**	7.36E-3	**2.19E-3**	7.81E-2	7.29E-3	2.41E-3	6.82E-2	4.10E-3	3.29E-3	7.45E-2
10-M-10	**1.21E-1**	**1.13E-1**	**3.41E-1**	1.80E-1	1.22E-1	4.02E-1	2.70E-1	1.87E-1	6.43E-1	3.07E-1	2.23E-1	6.96E-1
	1.20E-1	**4.16E-2**	**6.31E-2**	1.89E-1	5.46E-2	1.39E-1	3.68E-1	9.93E-2	1.56E-1	4.32E-1	1.16E-2	8.91E-2
10-M-20	**2.13E-2**	**1.02E-2**	**6.43E-1**	2.54E-2	1.20E-2	6.74E-1	3.06E-2	2.24E-2	7.29E-1	3.18E-2	3.07E-2	7.34E-1
	1.09E-2	**2.22E-3**	**4.13E-2**	**1.06E-2**	2.48E-3	9.89E-2	1.41E-2	2.89E-3	8.92E-2	8.76E-3	4.67E-3	8.37E-2
10-M-50	**1.55E-2**	1.86E-2	**4.63E-1**	1.75E-2	**1.75E-2**	7.70E-1	2.04E-2	1.98E-2	7.76E-1	2.11E-2	2.58E-2	8.23E-1
	5.08E-3	**2.97E-3**	**1.79E-2**	5.38E-3	3.24E-3	5.79E-2	8.60E-3	4.89E-3	2.62E-2	8.93E-3	6.33E-3	6.16E-2
10-L-10	**1.14E-1**	**7.50E-2**	**6.69E-1**	1.47E-1	1.37E-1	7.10E-1	1.94E-1	2.87E-1	7.88E-1	2.27E-1	3.43E-1	8.47E-1
	2.30E-2	**2.33E-2**	**1.12E-2**	7.09E-2	2.77E-2	8.04E-2	5.39E-2	6.30E-2	2.90E-2	5.04E-2	7.83E-2	4.80E-2
10-L-20	**2.52E-2**	**2.21E-2**	7.81E-1	2.75E-2	3.13E-2	**5.41E-1**	3.73E-2	3.08E-2	7.36E-1	3.79E-2	3.37E-2	6.92E-1
	6.36E-3	**3.11E-3**	**5.16E-2**	7.55E-3	1.37E-2	6.65E-2	1.54E-2	4.47E-3	1.13E-1	1.07E-2	7.03E-3	5.26E-2
10-L-50	**9.98E-3**	**1.51E-2**	**7.37E-1**	1.15E-2	1.52E-2	7.77E-1	1.51E-2	1.82E-2	7.70E-1	1.78E-2	2.06E-2	8.28E-1
	2.79E-3	**1.06E-3**	7.47E-2	2.94E-3	4.89E-3	**6.13E-2**	8.05E-3	4.77E-3	1.37E-1	1.05E-2	1.73E-3	9.14E-2
统计结果	**24/25**	**23/26**	**24/25**	**1/2**	**2/1**	**2/1**	**2/0**	**1/0**	**0/1**	**0/0**	**1/0**	**1/0**

From Table 4, the evaluation indicator values displayed in bold fonts are mostly concentrated on the running results of the SMOEA/D algorithm. SMOEA/D algorithm has obvious advantages in three evaluation indicators. On 27 instances, SMOEA/D algorithm has the best GD value and IGD value in all instances; According to the values of distributed evaluation indicator Spread, SMOEA/D algorithm obtained the best Spread value on 24 instances. Only in the instances of "2-M-20", "5-M-10" and "10-L-20", its performance was inferior to NSGAII algorithm and MOPSO algorithm. Among them, the Spread values on "2-M-20" and "10-L-20" were inferior to the other three comparative algorithms, and the NSGAII algorithm obtained the best Spread value; The MOPSO algorithm achieved the best Spread value on "10-L-20". Overall, SMOEA/D algorithm performs optimally in most of the instances and solves the collaborative scheduling problem of machining-assembly multi production lines when the number of processes in each production line is not equal.

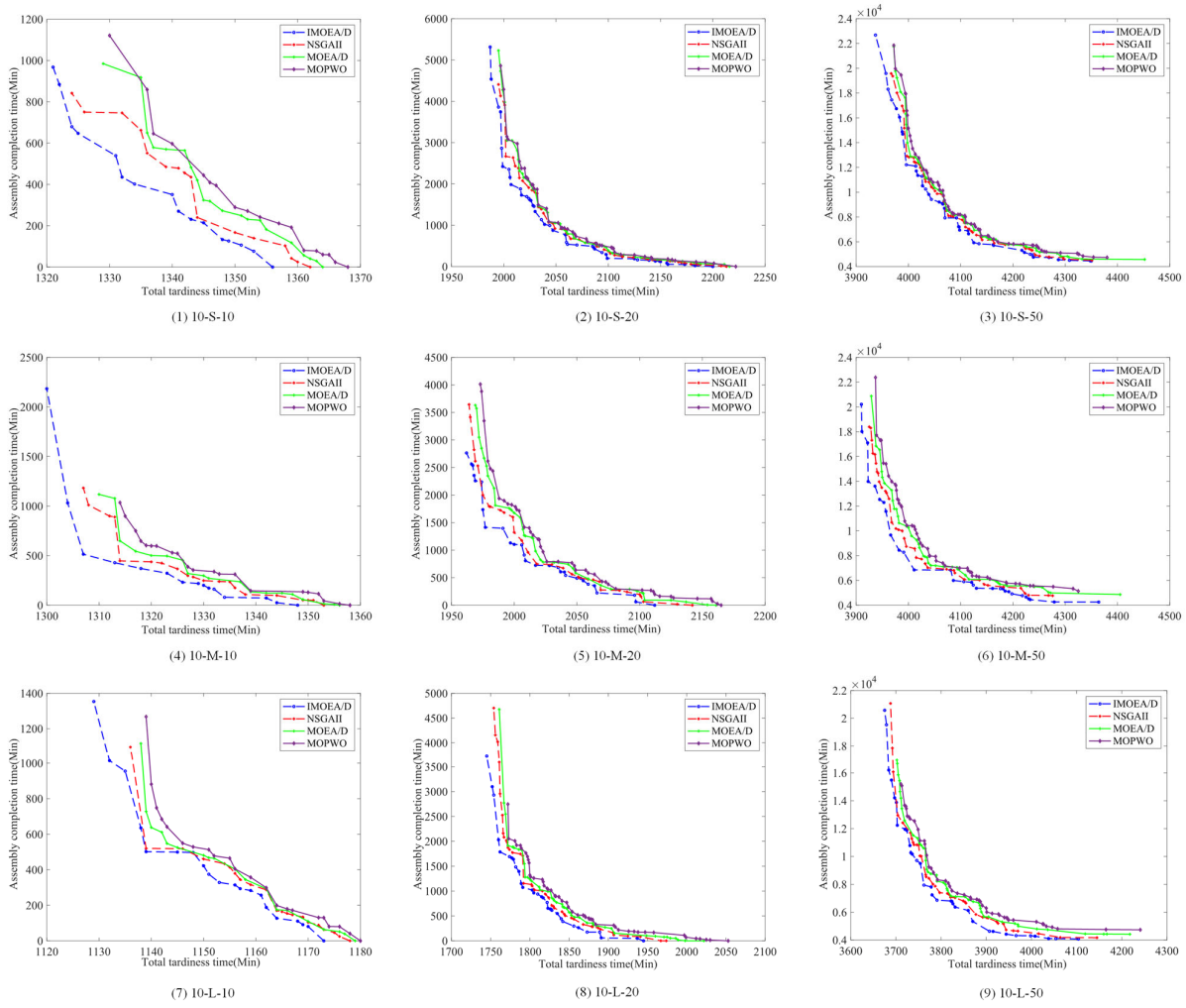

Fig. 6. Convergence curves of partial instances with the best IGD run by different MOEAs in experiment II

Fig. 6 shows the partial convergence curve when the IGD value is the best in 30 runs on partial instances when the number of operations of each production line is not equal. Overall, similar to the curve of Experiment I, the convergence curve corresponding to the SMOEA/D algorithm has better distribution in most instances. At the same time, as the size of the workpiece increases, the convergence curve of the algorithm is getting closer, but combined with the GD values of each test instance in Table 4, the convergence curve of SMOEA/D algorithm is still better than the other three algorithms.

From the analysis of two sets of experimental results and curve graphs, it can be seen that in solving the collaborative scheduling problem of machining-assembly multiple production lines, the SMOEA/D algorithm obtains better solutions than the other three comparative algorithms for the vast majority of instances, regardless of whether the number of operations for each production line is equal or the difference in the number of operations on each production line, for different problem scales; For the same instance, when the operations of the production line are equal or not, the optimization performance of the algorithm is not affected. Basically, a better solution can be obtained. Therefore, our proposed SMOEA/D algorithm is effective in solving the collaborative scheduling problem of multiple production lines and can meet the needs of problem solving.

6. Conclusion

This article investigates the collaborative scheduling problem of machining- assembly multiple production lines considering complete constraints. By analyzing the production characteristics of coach driving axle assembly, a mathematical model for collaborative scheduling of machining-assembly multiple production lines was constructed with the optimization objectives of minimizing Makespan and minimizing lead time penalties. In this problem model, the production process of the assembly line is constrained by the machining sequence of each subcomponent on the machining line. The assembly process of any type of product must start after the corresponding subcomponents on all machining lines have been processed. To solve this kind of problem, an enhanced MOEA/D algorithm (SMOEA/D) incorporating Tabu search strategy is proposed. Through the improvement of the coding method, parent individual selection strategy and the updating process of population, the global search ability and local search ability of the algorithm are improved, and the optimization performance of the algorithm is

improved. In the experimental design, in order to verify the effectiveness of the algorithm, two groups of experiments were designed, respectively considering the equal and unequal number of processes in each production line. The SMOEA/D algorithm and three other well-known multi-objective optimization algorithms were used to solve each test case. The experimental results show that SMOEA/D algorithm is superior to the other three comparison algorithms in solving the machine assembly multi production line collaborative scheduling problem in most test cases.

Acknowledgements

This work was supported by the International and Hong Kong, Macao and Taiwan high end talent exchange funding of Guangdong Province (No. 2022A0505020007), Guangzhou basic and applied basic research project (No. 202201010261).

References

Behnamian, J., & Ghomi, S. (2016). A survey of multi-factory scheduling. *Journal of Intelligent Manufacturing, 27*(1), 231-249.

Bhatnagar, R., Chandra, P., & Goyal, S. K. (1993). Models for multi-plant coordination. *European Journal of Operational Research, 67*(2), 141-160.

Chao, Lu, Liang, Gao, Xinyu, Li, et al. (2018). A multi-objective approach to welding shop scheduling for makespan, noise pollution and energy consumption. *Journal of Cleaner Production*.

Coello, C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 256-279.

Deb, K., & Jain, H. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation, 18*(4), 577-601.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182-197.

Deng, J., & Wang, L. (2017). A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. *Swarm and Evolutionary Computation, 32*, 121-131. <Go to ISI>://WOS:000392774200007

Hall, N. G., & Potts, C. N. (2003). Supply chain scheduling: batching and delivery. *Operations Research, 51*(4).

Hall, N. G., & Potts, C. N. (2005). The Coordination of Scheduling and Batch Deliveries. *Annals of operations research*, 135.

Ham, J. I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*.

Hao, L., Bing, D., Huang, G. Q., Chen, H., & Li, X. (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal Of Production Economics, 146*(2), 423–439.

Jiang, S., & Yang, S. (2017). A strength pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 329-346.

Jiang, S. W., Ong, Y. S., Zhang, J., & Feng, L. (2014). Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization. *IEEE Transactions on Cybernetics, 44*(12), 2391-2404.

Kuo, I. H., Horng, S. J., Kao, T. W., Lin, T. L., & Pan, Y. (2007). An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model. *Expert Systems with Applications, 36*(3), 7027-7032.

Li, H., & Zhang, Q. (2008). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE transactions on evolutionary computation*, *13*(2), 284-302.

Lu, C., Gao, L., Li, X., & Xiao, S. (2017). A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Engineering Applications of Artificial Intelligence, 57*, 61-79.

Mazdeh, M. M., Sarhadi, M., & Hindi, K. S. (2008). A branch-and-bound algorithm for single-machine scheduling with batch delivery and job release times. *Computers & Operations Research, 35*(4), 1099-1111.

Meng, R., Rao, Y., Zheng, Y., & Qi, D. (2017). Modelling and solving algorithm for two-stage scheduling of construction component manufacturing with machining and welding process. *International Journal of Production Research*(10), 1-13.

Pan, Q. K., Wang, L., Gao, L., & Li, W. D. (2011). An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers. *Information Sciences, 181*(3), 668-685.

Rahnamayan, S., Tizhoosh, H. R., & Salama, M. (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications, 53*(10), 1605-1614.

Sun, B., Zhai, G., Li, S., & Pei, B. (2023). Multi-resource collaborative scheduling problem of automated terminal considering the AGV charging effect under COVID-19. *Ocean & coastal management, 232*, 106422. <Go to ISI>://MEDLINE:36407122

Ullrich, & Christian, A. (2013). Integrated machine scheduling and vehicle routing with time windows. *European Journal of Operational Research, 227*(1), 152-165.

Wang, L., Pan, Q. K., & Tasgetiren, M. F. (2011). A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem. *Computers & Industrial Engineering, 61*(1), 76-83.

Yang, Z., Ma, Z., & Wu, S. (2016). Optimized flowshop scheduling of multiple production lines for precast production. *Automation in Construction*, *72*, 321-329.

Yue, L., Guan, Z., Zhang, L., Ullah, S., & Cui, Y. (2019). Multi objective lotsizing and scheduling with material constraints

in flexible parallel lines using a Pareto based guided artificial bee colony algorithm. *Computers & Industrial Engineering, 128*(FEB.), 659-680.

Zhang, Q., & Hui, L. (2008). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation, 11*(6), 712-731.

Zhang, Q., Hui, L., Maringer, D., & Tsang, E. (2010). *MOEA/D with NBI-style Tchebycheff approach for portfolio management.* Paper presented at the Evolutionary Computation.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions On Evolutionary Computation, 7*(2), 117-132.