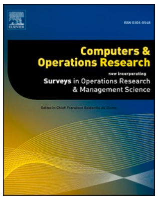
Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor



A hybrid collaborative framework for integrated production scheduling and vehicle routing problem with batch manufacturing and soft time windows

Ming Huang[a,b], Baigang Du[a,b], Jun Guo[a,b,*]

[a] *School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China*
[b] *Hubei Digital Manufacturing Key Laboratory, Wuhan 430070, China*

ARTICLE INFO

Keywords:
Scheduling
Integrated production-distribution
Batch manufacturing
Soft time windows
Hybrid collaborative framework

ABSTRACT

This paper studies a new integrated production scheduling and vehicle routing problem where the production of customer orders is performed under a batch manufacturing environment and order deliveries are made by multi-trip heterogeneous vehicles in soft time windows. A bi-objective mixed-integer programming model with maximizing total profits and minimizing total weighted earliness and tardiness has been established. We develop a hybrid collaborative framework to solve this problem, which nests the collaborative mechanism in an optimization mode based on the hybrid algorithm. In the collaborative mechanism, a property on the ideal optimal departure time of the tour is first proposed, based on which an exact strategy is developed to simultaneously coordinate batch manufacturing and tour departure schedules. High-quality integrated solutions are provided by simultaneously making both production scheduling and vehicle routing decisions. Then, in order to get the best integrated solution, we adopt a multi-objective evolutionary algorithm improved by an adaptive large neighborhood search strategy based on the specific problem and coding form to realize the optimization mode. Computational experiments are performed on a dataset containing 30 instances of various scales. The results show that the proposed hybrid collaborative framework performs well in cardinality, convergence, distribution and spread, which is a very competitive method to solve this problem.

1. Introduction

Production and distribution are important issues in supply chain management. Especially in fast-moving consumer goods (FMCG) (e.g., beverages, bread, etc.) manufacturing enterprises, products usually have features such as short fixed shelf lifespan and rapid circulation (Liu et al., 2021). Timely delivery of products is required after being manufactured, thereby a rational production and distribution plan is crucial to enterprise operation. The traditional decision of production and distribution is performed separately and sequentially, which guarantees timely delivery by way of premature manufacturing or frequent vehicle deliveries (Fu et al., 2017). However, these two ways will lead to high inventory costs or distribution costs, and it is difficult to ensure effective coordination of production and distribution (Absi et al., 2018; Darvish and Coelho, 2018). Therefore, in order to reduce the overall cost of the supply chain while satisfying customer expectations for timely delivery, it is necessary to design a coordinated integrated production and distribution plan for FMCG manufacturing enterprises.

With the increasingly fierce market competition, customers put forward higher requirements for timely delivery. Due to the difference in order delivery time and the limitations of production line capacity, the quantity and capacity of delivery vehicles, it is difficult to force all orders to be delivered within the specified time window (Figliozzi, 2010). Thus, it is reasonable and necessary to consider orders with soft time windows that allow delivery beyond the time window but both early and tardy delivery degrade customer satisfaction (Oladzad-Abbasabady et al., 2023; Zhao et al., 2023). In addition, the consideration of soft time windows provides flexibility in the departure time of the vehicle tour (Mohammadi et al., 2020). The flexible departure time of the tour means that the tour can be delivered at any moment after all orders assigned to this tour have been produced, but different departure times correspond to different earliness or tardiness for orders on this tour.

In many real-life FMCG manufacturing enterprises, batch manufacturing becomes a common practice for reducing the inventory backlog. And mass production with large-scale inventory is no longer a desirable decision (Vahdani et al., 2017). A batch is defined as a group of orders consisting of different varieties of products assigned to a vehicle tour. There is a one-to-one correspondence between batches and tours.

* Corresponding author.
E-mail addresses: huangming@whut.edu.cn (M. Huang), junguo@whut.edu.cn (J. Guo).

https://doi.org/10.1016/j.cor.2023.106346
Received 7 October 2022; Received in revised form 27 June 2023; Accepted 2 July 2023
Available online 3 July 2023

Table 1
Summary of the literature review.

Author(s)	Production characteristics		Distribution characteristics						Optimization objectives	
			Delivery time constraints			Vehicle configurations				
	Batch manufacturing	Multiple products	Soft time window	Hard time window	Delivery due date	Vehicle number limit	Heterogeneous vehicle	Multiple trips	Cost/Profit	Customer satisfaction
(Devapriya et al., 2017)	●				●	●		●	DC	
(Kergosien et al., 2017)		●			●	●		●		TP
(Guo et al., 2017)	●	●			●		●		PC + IC + DC	TP
(Chevroton et al., 2021)	●				●				IC + DC	TP
(Yagmur and Kesen, 2021)	●				●	●	●	●		TP
(Yagmur and Kesen, 2023)					●	●	●	●		TP
(Fu et al., 2017)		●		●			●		PC + DC	
(Vahdani et al., 2017)		●		●		●	●	●	PC + IC + DC	
(Park and Hong, 2009)		●		●					PC + DC	TP
(Wang et al., 2019)				●		●		●	DC	TP
(Ganji et al., 2020)	●			●		●	●	●	DC	TP
(Hou et al., 2022)			●			●				EP + TP
(Mohammadi et al., 2020)		●	●			●	●		PC + DC	EP + TP
(Liu et al., 2021)			●			●	●		DC	EP + TP
Current study	●	●	●			●	●	●	PC + IC + DC	EP + TP

PC: Production cost; IC: Inventory cost; DC: Distribution cost; EP: Early penalty; TP: Tardy penalty.

In addition, batch manufacturing and vehicle tours with flexible departure times make the formulation of an integrated manufacturing-inventory-vehicle routing plan challenging. We not only need to conduct a rational tour (batch) division, but also to determine the production time of each batch and the tour departure time based on reducing inventory costs and meeting the delivery time constraints for customers as possible. And batch manufacturing is strongly associated with vehicle tour departure schedules. Therefore, it is important to establish a collaborative mechanism for production scheduling with batch manufacturing and vehicle routing with flexible departure time in the manufacturing-inventory-vehicle routing integration.

Our research can be regarded as a novel variation of an integrated production-distribution scheduling (IPDS) problem, which concerns batch manufacturing and soft time windows. We aim to find an integrated production scheduling and vehicle routing solution, while achieving maximum total profits and the highest possible customer satisfaction. The main contribution is as follows:

- We consider a novel production-distribution environment involving production scheduling with batch manufacturing and vehicle routing with soft time windows and multi-trip heterogeneous vehicles. This problem is investigated for the first time in the related literature.
- A bi-objective mixed-integer programming model with maximizing total profits and minimizing total weighted earliness and tardiness is established. A hybrid collaborative framework is designed to solve this model, which includes a collaborative mechanism and an optimization mode based on the hybrid algorithm.
- In the collaborative mechanism, according to the delivery penalty time function, the ideal optimal departure time of the tour is first proposed, based on which an exact strategy is developed to simultaneously coordinate batch manufacturing and tour departure schedules.

This paper is organized as follows: Section 2 reviews relevant literature about IPDS. Section 3 gives a formal description of the problem and a mixed-integer programming model. Section 4 details the design and realization of the hybrid collaboration framework. Section 5 proves the applicability of our model through 30 instances of various scales, and verifies the superiority of the proposed framework. Finally, Section 6 gives conclusions and future research directions.

2. Literature review

In recent years, the IPDS problem involving routing decisions has received increasing attention from researchers. For interested readers, the study of Moons et al. (2017) can be referred. Table 1 provides an overview of related research to clearly recognize the value of our research.

For production characteristics, most publications regard an order as a job, and production scheduling is carried out for each job (Fu et al., 2017; Kergosien et al., 2017; Yagmur and Kesen, 2023). Single product variety (Chevroton et al., 2021; Devapriya et al., 2017; Ganji et al., 2020; Yagmur and Kesen, 2021). More specifically, very few studies on IPDS looked at the production characteristics of multi-variety products batch manufacturing. Although Guo et al. (2017) considered the multi-product batch manufacturing model, they assumed only continuous manufacturing between batches.

For delivery characteristics, some publications involved delivery due dates (Devapriya et al., 2017; Guo et al., 2017; Kergosien et al., 2017; Yagmur and Kesen, 2021; Yagmur and Kesen, 2023). As the research progressed, realistic orders tended to be required to be delivered within a time interval, thereby some studies proposed hard time windows. Fu et al. (2017), and Vahdani et al. (2017) considered a hard time window with compulsory constraints at both ends, and orders must be delivered within this time window. Some publications involved hard time windows with compulsory constraints at one end, orders can be delivered early, but tardiness cannot exceed a hard upper limit of the time window (Ganji et al., 2020; Park and Hong, 2009). Wang et al. (2019) considered the situation where the vehicle needs to wait until the hard lower limit of the time window to execute delivery when the vehicle arrives early. Due to hard time windows with compulsory constraints being ideal, soft time windows with flexible delivery properties are gradually being studied. For the relevant literature involving soft time windows, most of the literature deals with the integration problem through a sequential coordination method. Among them, Hou et al. (2022) proposed a coordination method that first conducts job assignments to obtain the production sequence, and then determines the delivery plan based on the production sequence and homogeneous vehicle capacity. Mohammadi et al. (2020) coordinated production and delivery by fixing the departure time of the vehicle tour. It is assumed that the starting time of each vehicle's tour is equal to the completion time of the last job in the vehicle's cargo. Thus, vehicle tour departure times can be determined by production schedules. The coordination between them is simplified. Liu et al. (2021) considered the feature of flexible departure time of the tour.

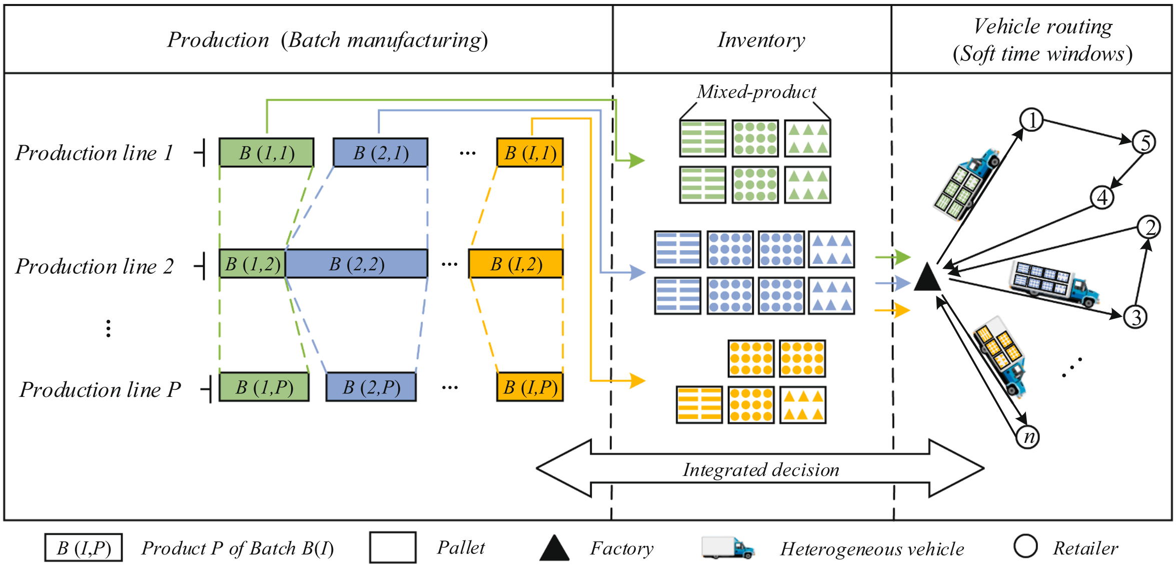


Fig. 1. Illustration of an integrated production scheduling and vehicle routing.

But the proposed coordination method was to first obtain the shop scheduling scheme with the minimum makespan, and then an exact procedure was used to determine the optimal tour departure time of this shop scheduling scheme to form the integrated solution. This method was to coordinate production and vehicle tour departures sequentially rather than simultaneously. To the best of our knowledge, there is no paper considered a collaborative mechanism that simultaneously coordinates production scheduling and vehicle routing.

Consequently, this study is among the first investigations studying the simultaneous coordination of batch manufacturing and tour departure schedules. In addition, according to Table 1, a bi-objective model with maximizing total profits (Including production, inventory, and distribution costs) and minimizing total weighted earliness and tardiness is rarely investigated.

3. Problem description and mathematical formulation

3.1. Problem statement and assumptions

The investigated IPDS problem includes the production stage with batch manufacturing, inventory stage, and distribution stage (vehicle routing) with soft time windows. As shown in Fig. 1, the production stage consists of several independent production lines. Each production line produces a single variety of products. Each batch contains multiple varieties of products, where $B(I, P)$ represents product P of batch $B(I)$. The production line needs to shut down and re-preparation when there is an idle time between adjacent batches (e.g., $B(1, 1)$ and $B(2, 1)$), thereby incurring production start-up costs. The batch can only be delivered when all the products in the batch have been produced. Moreover, each variety of product within each batch will incur inventory if it is not delivered immediately after production. The factory has enough inventory capacity, and the storage and delivery of products are based on pallets. The distribution stage involves a vehicle routing problem. It is defined on the entire graph $G = (N, A)$, which contains various vertices $N = \{0, 1, \ldots, n\}$ and the arc set is $A = \{(k, l){:}\ k, l \in N, k<l\}$. The factory is placed at vertex 0, and the set of retailers is described as $R = N\backslash\{0\}$. A limited number of multi-trip heterogeneous vehicles are used to deliver products to a set of geographically dispersed retailers. Earliness and tardiness in deliveries result in a drop in customer satisfaction. Each vehicle starts its route from the factory,

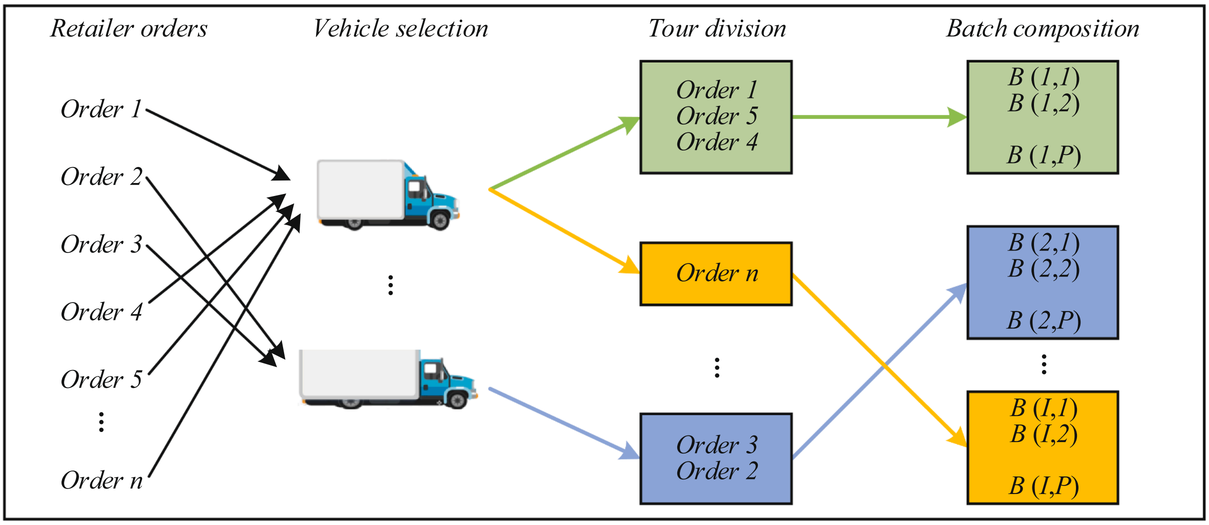


Fig. 2. Flowchart for breaking orders into batches.

visits a sequence of retailers and returns to the factory.

Integrated decision-making includes breaking orders into batches and batch manufacturing-inventory-vehicle routing. Fig. 2 shows a flowchart for breaking orders into batches. First, we select the delivery vehicle for the order according to the size of the order. Then, the orders assigned to each vehicle are divided into different vehicle tours based on vehicle capacity and soft time windows. And the products in a tour compose a batch. Each tour has a flexible departure time, but different departure times correspond to different early or tardy penalty times for orders on this tour. Finally, the manufacturing sequence of batches is obtained by calculating the ideal optimal departure time for each tour.

Subsequently, the integrated production scheduling and vehicle routing solution is obtained by simultaneously coordinating batches with manufacturing sequences and tours with ideal optimal departure times. Noticeably, when the production of a certain variety of products in a batch (i.e., $B(I,P)$) is completed, the products need to be packed into pallets and transported to the inventory for storage. Therefore, the inventory time of different varieties of products within a batch may not be the same. The delivery can only start once all varieties of products within a batch have been produced and packed into pallets.

Some detailed assumptions for this problem are:

- The production process within the batch is continuous on each production line.
- Each retailer's demand must be satisfied in one delivery, and separate deliveries are not allowed.
- The total demand for each order must not exceed the capacity of the vehicles.
- Delivery is allowed for vehicles that are not fully loaded, and products are allowed to be mixed.
- During each tour, the vehicle is delivered continuously without waiting.

3.2. Notations of parameters

The important notations used in formulating the mathematical model of the investigated IPDS problem are listed below. Where the unit of price and cost is yuan and the unit of time is the hour.

Indices

p: Index for product variety (production line) ($p = (1, 2, \cdots\cdots P)$).

i,j: Index for the production sequence of batches ($i,j = (1, 2, \cdots\cdots n)$).

k, l: Indices for the retailers ($k, l = (1, 2, \cdots\cdots n)$).

$0, n+1$: The factory index at the starting and the final location.

v: Index for the vehicle ($v = (1, 2, \cdots\cdots V)$).

Constants

$D_{k,p}$: The demand of retailer k for product p.

Pr_p: The average sale price of product p.

Pc_p: The average production cost of product p.

Pt_p: Processing time of product p.

Fs_p: The production start-up cost of product line p.

O_p: Unit product p occupies the capacity per pallet.

Hc: Holding cost rate of inventory per pallet product.

C_v: Maximum load number of pallets for vehicle v.

Fc_v: Fixed cost of vehicle v.

Vc_v: The variable cost of vehicle v per unit time.

$T_{k,l}$: Travel time from the location of retailer k to the location of retailer l.

$[E_k, L_k]$: The time window of retailer k.

Re: Early time penalty rate for per pallet product.

Rd: Tardy time penalty rate for per pallet product.

M: A sufficiently large number.

Intermediate variables

$tc_{p,B(i)}$: Completion time of product p for production batch $B(i)$.

a_k: Arrival time of vehicle delivery to retailer k.

$s_{B(i),B(j)}$: 1, if batch $B(i)$ is immediately followed by batch $B(j)$ during a vehicle; 0, otherwise.

$w_{p,B(i)}$: 1, if $ts_{p,B(i+1)} > tc_{p,B(i)}$; 0, otherwise.

Decision variables

$ts_{p,B(i)}$: Start time of product p for production batch $B(i)$.

$td_{B(i)}$: Departure time of production batch $B(i)$.

$e_{B(i)}$: 1, batch $B(i)$ exists; 0, otherwise.

$x_{k,l}$: 1, if retailer k is immediately followed by retailer l during a vehicle's tour (batch); 0, otherwise.

$y_{v,B(i)}$: 1, if the production batch $B(i)$ is delivered by vehicle v; 0, otherwise.

$g_{k,B(i)}$: 1, if the order of retailer k is included in product batch $B(i)$; 0, otherwise.

3.3. A mixed-integer programming model

The mathematical model for the investigated IPDS problem is formulated. The specific descriptions of the two objective functions and constraints are as follows:

(1) Total profits.

Total profits consist of product revenue, production start-up cost, inventory cost, and distribution cost. The production start-up cost is directly related to the number of idle occurrences. And inventory cost is related to the number of pallets occupied by the product and storage time. In addition, the delivery cost consists of two parts. One is a fixed cost related to the number of vehicle departures, and the other is a variable cost related to vehicle routing time (fuel consumption).

$$\begin{aligned}\text{Max}\ &\sum_{p=1}^{P}\sum_{k=1}^{n}\left(Pr_p - Pc_p\right)D_{k,p} - \sum_{p=1}^{P}\sum_{i=1}^{n-1} w_{p,B(i)}Fs_p \\ &-Hc\sum_{i=1}^{n}\sum_{p=1}^{P}\sum_{k=1}^{n}\left[g_{k,B(i)}D_{k,p}O_p\left(td_{B(i)} - tc_{p,B(i)}\right)\right] \\ &-\sum_{v=1}^{V}\left\{Fc_v\sum_{i=1}^{n}y_{v,B(i)} + Vc_v\sum_{k=1}^{n}\sum_{i=1}^{n}\left[g_{k,B(i)}y_{v,B(i)}\left(x_{0,k}T_{0,k} + \sum_{l=1}^{n+1}x_{k,l}T_{k,l}\right)\right]\right\}\end{aligned} \tag{1}$$

(2) Customer satisfaction.

Considering the delivery time constraints of orders, timely delivery within soft time windows helps to improve customer satisfaction. This paper measures customer satisfaction objective by total weighted earliness and tardiness. It is determined by the early or tardy time of the delivery order, the number of pallets occupied by the order and the penalty rate.

$$\text{Min}\ \sum_{k=1}^{n}\sum_{p=1}^{P}\left[D_{k,p}O_p\left(Re\max(E_k - a_k, 0) + Rd\max(a_k - L_k, 0)\right)\right] \tag{2}$$

Subject to:

$$ts_{p,B(i)} + Pt_p\sum_{k=1}^{n}g_{k,B(i)}D_{k,p} = tc_{p,B(i)} \quad \forall\, p \in \{1, \cdots, P\},\ \forall\, i \in \{1, \cdots, n\} \tag{3}$$

$$tc_{p,B(i)} \leqslant ts_{p,B(i+1)} + M\left(1 - e_{B(i+1)}\right) \quad \forall\, p \in \{1, \cdots, P\},\ \forall\, i \in \{1, \cdots, n-1\} \tag{4}$$

$$tc_{p,B(i)} \leqslant td_{B(i)}\ \forall\, p \in \{1, \cdots, P\},\ \forall\, i \in \{1, \cdots, n\} \tag{5}$$

$$td_{B(i)} = \sum_{k=1}^{n}\left(a_k - T_{0,k}\right)x_{0,k}g_{k,B(i)}\ \forall\, i \in \{1, \cdots, n\} \tag{6}$$

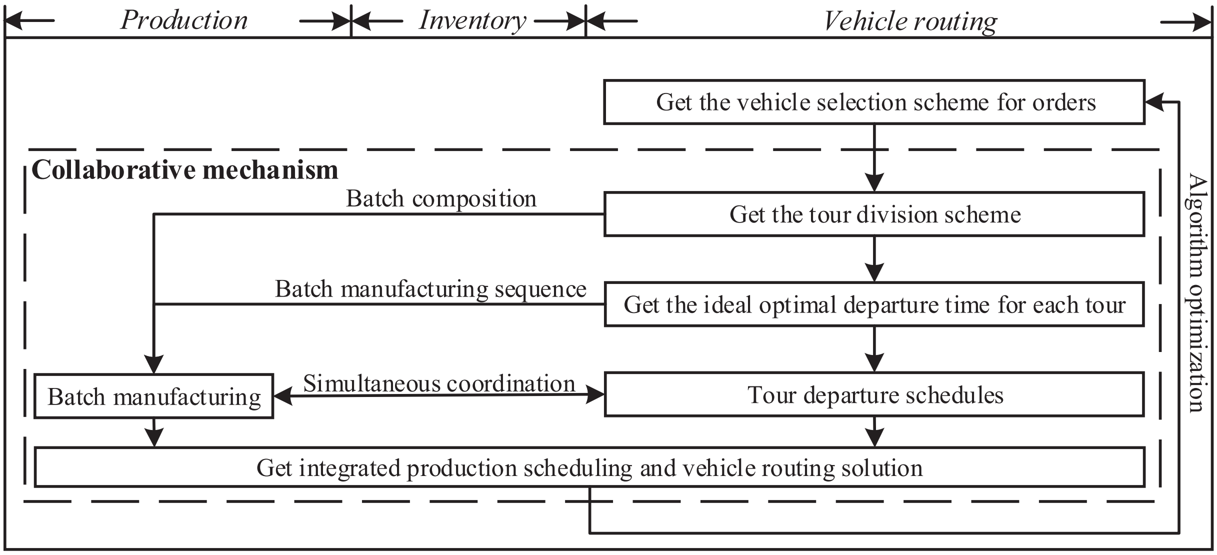

Production
Inventory
Vehicle routing
Get the vehicle selection scheme for orders
Collaborative mechanism
Batch composition
Get the tour division scheme
Batch manufacturing sequence
Get the ideal optimal departure time for each tour
Batch manufacturing
Simultaneous coordination
Tour departure schedules
Get integrated production scheduling and vehicle routing solution
Algorithm optimization

Fig. 3. Logic diagram of hybrid collaborative framework.

$$\sum_{l=1}^{n} x_{k,l}\left(a_k + T_{k,l} - a_l\right) = 0\ \forall\ k \in \{1,\cdots,n\} \tag{7}$$

$$\sum_{l=1}^{n}\left(a_l + T_{l,n+1}\right)x_{l,n+1}g_{l,B(i)} \leqslant td_{B(j)} + M\left(1 - s_{B(i),B(j)}\right) \quad \forall\ i \in \{1,\cdots,n-1\},\ \forall\ j \in \{i+1,\cdots,n\} \tag{8}$$

$$\sum_{l=1}^{n} x_{0,l} = \sum_{k=1}^{n} x_{k,n+1} = \sum_{i=1}^{n} e_{B(i)} \tag{9}$$

$$\sum_{k=0}^{n} x_{k,l} = 1\ \forall\ l \in \{1,\cdots,n\} \tag{10}$$

$$\sum_{l=1}^{n+1} x_{k,l} = 1\ \forall\ k \in \{1,\cdots,n\} \tag{11}$$

$$E_k \leqslant E_l + M\left(1 - x_{k,l}\right) \quad \forall\ k \in \{1,\cdots,n\},\ \forall\ l \in \{1,\cdots,n\} \tag{12}$$

$$L_k \leqslant L_l + M\left[1 + x_{k,l}(E_l - E_k - 1)\right] \quad \forall\ k \in \{1,\cdots,n\},\ \forall\ l \in \{1,\cdots,n\} \tag{13}$$

$$\sum_{i=1}^{n} g_{k,B(i)} = 1\ \forall\ k \in \{1,\cdots,n\} \tag{14}$$

$$\sum_{v=1}^{V} y_{v,B(i)} \leqslant 1\ \forall\ i \in \{1,\cdots,n\} \tag{15}$$

$$1 - M\left(1 - e_{B(i)}\right) \leqslant \sum_{k=1}^{n} g_{k,B(i)} \leqslant Me_{B(i)}\ \forall\ i \in \{1,\cdots,n\} \tag{16}$$

$$1 - M\left(1 - e_{B(i)}\right) \leqslant \sum_{v=1}^{V} y_{v,B(i)} \leqslant Me_{B(i)}\ \forall\ i \in \{1,\cdots,n\} \tag{17}$$

$$g_{k,B(i)} \geqslant g_{l,B(i)} + x_{k,l} - 1 \quad \forall\ k \in \{1,\cdots,n\},\ \forall\ l \in \{1,\cdots,n\},\ \forall\ i \in \{1,\cdots,n\} \tag{18}$$

$$g_{l,B(i)} \geqslant g_{k,B(i)} + x_{k,l} - 1 \quad \forall\ k \in \{1,\cdots,n\},\ \forall\ l \in \{1,\cdots,n\},\ \forall\ i \in \{1,\cdots,n\} \tag{19}$$

$$E_{B(i+1)} \leqslant ME_{B(i)}\ \forall\ i \in \{1,\cdots,n-1\} \tag{20}$$

$$\sum_{p=1}^{P}\sum_{k=1}^{n} g_{k,B(i)}D_{k,p}O_p \leqslant C_v + M\left(1 - y_{v,B(i)}\right) \quad \forall\ v \in \{1,\cdots,V\},\ \forall\ i \in \{1,\cdots,n\} \tag{21}$$

$$e_{B(i)}, x_{k,l}, y_{v,B(i)}, g_{k,B(i)}, s_{B(i),B(j)}, w_{p,B(i)} \in \{0,1\} \tag{22}$$

$$ts_{p,B(i)}, tc_{p,B(i)}, td_{B(i)}, a_k \geqslant 0 \tag{23}$$

Constraints (3)-(4) indicates the time constraint of batches on production lines. Constraint (5) indicates that a batch of products can be delivered only after all products are completed. Constraints (6)-(13) are descriptions of constraints on routing time and sequence. Constraint (6) indicates the time constraint from the factory to the first retailer. Constraint (7) indicates the constraints of the delivery sequence and time of the two retailers during a batch. Constraint (8) indicates the vehicle can't deliver the next batch until it completes the current delivery task and returns to the factory. Constraint (9) indicates the tour takes the factory as the start and final node, tour and batch one-to-one correspondence. Constraints (10)-(11) ensure that the retailer is only visited once and it is only departed once in the routing process. Constraints (12)-(13) indicate the visited sequence of retailers on each tour. Constraints (14)-(15) indicate that a retailer's order can only exist in a batch and a batch can only be delivered by one vehicle. Constraint (16) indicates that the batch can only contain orders if the batch exists. Constraint (17) indicates that the batch can only be delivered by the vehicle if the batch exists. Constraints (18)-(19) indicate that retailer *k* is followed by retailer *l* during a tour. If retailer *k* is delivered by vehicle *v* then retailer *l* will also be delivered by vehicle *v* and vice versa. Constraint (20) ensures that a batch exists only if the previous batch exists. Constraint (21) indicates that the capacity occupied by the batch should not exceed the capacity of the delivery vehicle. Finally, constraints (22)-(23) define the domain and nature of the variables.

4. The hybrid collaborative framework

4.1. Design of the framework

As shown in Fig. 3, a hybrid collaborative framework is proposed to solve this problem, which includes a collaborative mechanism and an optimization mode based on the hybrid algorithm. The collaborative mechanism is designed as follows. Firstly, the vehicle selection scheme for orders is obtained, and the orders assigned to each vehicle are divided into different tours. The products in a tour compose a batch.

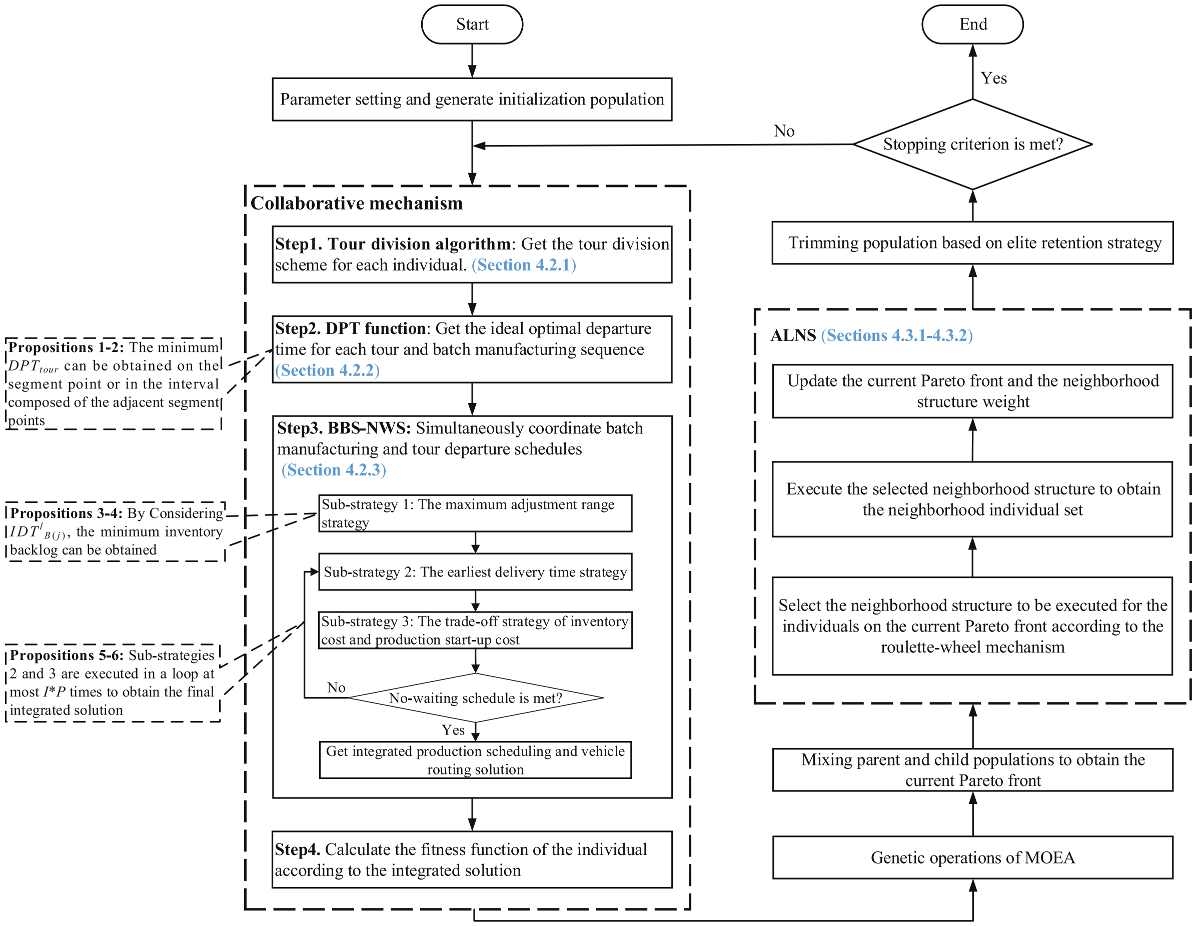

Start
Parameter setting and generate initialization population
Collaborative mechanism
Step1. Tour division algorithm: Get the tour division scheme for each individual. (Section 4.2.1)
Step2. DPT function: Get the ideal optimal departure time for each tour and batch manufacturing sequence (Section 4.2.2)
Step3. BBS-NWS: Simultaneously coordinate batch manufacturing and tour departure schedules (Section 4.2.3)
Sub-strategy 1: The maximum adjustment range strategy
Sub-strategy 2: The earliest delivery time strategy
Sub-strategy 3: The trade-off strategy of inventory cost and production start-up cost
No
No-waiting schedule is met?
Yes
Get integrated production scheduling and vehicle routing solution
Step4. Calculate the fitness function of the individual according to the integrated solution
Propositions 1-2: The minimum DPT_{tour} can be obtained on the segment point or in the interval composed of the adjacent segment points
Propositions 3-4: By Considering $IDT^l_{B(j)}$, the minimum inventory backlog can be obtained
Propositions 5-6: Sub-strategies 2 and 3 are executed in a loop at most $I*P$ times to obtain the final integrated solution
End
Yes
No
Stopping criterion is met?
Trimming population based on elite retention strategy
ALNS (Sections 4.3.1-4.3.2)
Update the current Pareto front and the neighborhood structure weight
Execute the selected neighborhood structure to obtain the neighborhood individual set
Select the neighborhood structure to be executed for the individuals on the current Pareto front according to the roulette-wheel mechanism
Mixing parent and child populations to obtain the current Pareto front
Genetic operations of MOEA

Fig. 4. Realization of the hybrid collaborative framework —IMOEA.

Secondly, the ideal optimal departure time for each tour is calculated based on the soft time window of orders, thereby obtaining the batch manufacturing sequence. Finally, batch manufacturing and tour departure schedules are simultaneously coordinated to determine the integrated production scheduling and vehicle routing solution. The algorithm optimization mode aims to obtain the best integrated production scheduling and vehicle routing solution by optimizing vehicle selection schemes.

Based on the logical idea in Fig. 3, an improved multi-objective evolutionary algorithm (IMOEA) is proposed to realize this framework shown in Fig. 4. First, a one-dimensional integer coding vector is used to represent the individual. The individual is composed of n code positions (number of retailers), and each code position stores the vehicle serial number ($1,\cdots, V$) selected by the retailer. Each individual represents a vehicle selection scheme. Then, the initial population is randomly generated to ensure its diversity. In addition, the fitness function of individuals is calculated based on the collaborative mechanism realization process (Step1-Step4). The tour division algorithm is provided in Section 4.2.1. We present the delivery penalty time (DPT) function to exactly obtain the ideal optimal departure time for each tour in Section 4.2.2. Especially for the simultaneous coordination problem, we specially developed a bidirectional scheduling strategy based on a no-wait schedule (BSS-NWS) to solve it in Section 4.2.3. Furthermore, we adopt a multi-objective evolutionary algorithm (MOEA) improved by ALNS based on the specific problem and coding form to realize the hybrid algorithm optimization mode (Sections 4.3.1–4.3.2). More specifically, MOEA is designed based on NSGA-II (Deb et al., 2002). The genetic operations of MOEA include selection, crossover, and mutation operations. The selection and crossover operations follow binary tournament selection and standard two-point crossover in NSGA-II. We specially designed a multi-point mutation operation based on small probability, and the number of mutation code positions is also random. It aims at randomly changing delivery vehicles for some retailers and expanding the search for vehicle selection schemes. Moreover, the Pareto front in each iteration is obtained by the non-dominated sorting method and crowding distance assignment method in NSGA-II. Finally, the individuals on the Pareto front are selected as the best solution set when the stopping criterion is met.

4.2. Realization of collaborative mechanism

4.2.1. Tour division algorithm

To guarantee a high service level to retailers, we focus on the influence of soft time windows in the design of the tour division algorithm. A tour division algorithm based on vehicle capacity and soft time windows is described in Algorithm 1.

Algorithm 1 Tour division algorithm.

```
$N_v$: The number of retailers delivering in the same vehicle.
$O_c$: The cumulative number of pallets occupied by the current tour.
$C_v$: Maximum load number of pallets for vehicle $v$.
$RL$ is a list of retailers sorted by the time window $[E_k, L_k]$ of each retailer in non-
    decreasing order.
For $i \in \{1, \cdots, N_v\}$ do
    Update $O_c$;
    If $O_c > C_v$ then
    Put $RL(i)$ in the next tour; Update $O_c$;
```

(continued on next page)

(*continued*)

```
Else
    If $i = 1 or N_v$ then
        Put $RL(i)$ in the current tour; Update $O_c$;
    Else
        $L_k$ of the first retailer $k$ in the current tour is used as the starting time to calculate $tc_l$,$tn_e$.
        $tc_l$: The latest arrival time of the vehicle when the retailer $RL(i)$ is placed in the current tour.
        $tn_e$: The earliest arrival time of the vehicle when the retailer $RL(i)$ is placed in the next tour.
        If $tc_l < E_{RL(i)}$ and $tn_e < L_{RL(i)}$ then
            Put $RL(i)$ in the next tour; Update $O_c$;
        Else
            Put $RL(i)$ in the current tour; Update $O_c$;
        End if
    End if
End if
End for
Get the tour division scheme.
```

4.2.2. *Determining the ideal optimal departure time for each tour: DPT function*

DPT is the penalty time for the delivery order beyond the time window. For a certain retailer, if the delivery vehicle arrives early or tardy, DPT will increase linearly as the time gap increases. When the arrival time is within the time window, DPT=0. Therefore, the DPT function is piecewise linear as shown in Fig. 5.

The notations are the following:

- t represents the abscissa, indicating the departure time of the tour.
- DPT represents the ordinate, each t corresponds to a delivery penalty time.
- t_1, t_2 are the segmentation points of the Piecewise linear function and the departure time of the tour corresponding to the retailer's time window.
- c_1, c_2 represent the y-intercept coefficients of each segment, respectively.
- k_1, k_2 represent the slope coefficients of each segment, respectively.

Notice that this slope is related to the number of retailers' products and the early or tardy penalty rate.

A tour contains a set of retailers, DPT_{tour} represents the total penalty time of a tour, which is composed of the sum of many piecewise linear functions. t corresponding to the minimized DPT_{tour} is the ideal optimal departure time. Assuming that there are m retailers in a tour, we can get at most $2m$ segment points $\{t_1, \cdots, t_{2m}\}$, dividing the abscissa into $2m$+1 segment intervals.

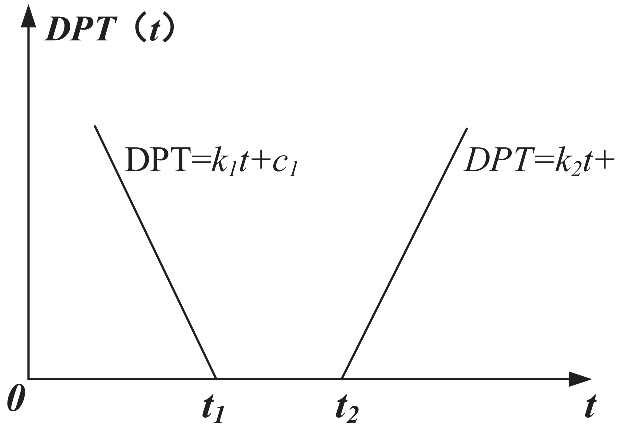


Fig. 5. Retailer's DPT function.

Proposition 1. *DPT_{tour} is a piecewise linear function.*

Proof. In a tour, multiple retailers will generate multiple piecewise linear functions. Since $(t_i, t_{i+1}) i \in \{0, \cdots, 2m\} t_0 = 0; t_{2m+1} = +\infty$ is the smallest segmentation interval, the DPT function of any retailer on (t_i, t_{i+1}) is linear. The linear function is still a linear function after linear operations. So DPT_{tour} is a linear function in each interval (t_i, t_{i+1}), thereby DPT_{tour} is a piecewise linear function in the entire interval $(0, +\infty)$. QED.

Proposition 2. *The departure time t corresponding to the minimized DPT_{tour} is obtained at the segment point. If the adjacent segment points simultaneously obtain the minimum DPT_{tour}, then the minimum DPT_{tour} can be obtained in the interval composed of the adjacent segment points.*

Proof. From the nature of the continuous function on the closed interval, we know that the minimized DPT_{tour} must exist in $[t_1, t_{2m}]$, and it can only be obtained at the terminal points, stationary points, and non-differentiable points. Obviously, for piecewise linear functions, the non-differentiable point must be the piecewise point. Since DPT_{tour} is linear in every interval $(t_i, t_{i+1}) i \in \{1, \cdots, 2m - 1\}$, if a stationary point exists, it must always exist in an interval with the segment point as the terminal point. Therefore, we only need to find the DPT_{tour} function value at each segment point and compare the values to find the minimized DPT_{tour}. It is worth noting that if the adjacent segment points $\{t_i, t_{i+1}\}$ simultaneously obtain the minimized DPT_{tour}, then due to the linear nature of DPT_{tour} in the interval $[t_i, t_{i+1}]$, the minimized DPT_{tour} is obtained in the interval $[t_i, t_{i+1}]$. QED.

Therefore, the ideal optimal departure time node (or time period) for each tour (batch) can be found by the DPT function without calculating all nodes in the interval $[t_1, t_{2m}]$. According to the ideal optimal departure time, the tours are sorted non-decreasing to obtain the manufacturing sequence of the corresponding batches. Algorithm 2 describes this solving process, where $Seq = \{Seq(1), \cdots Seq(h), \cdots Seq(H)\}$ represents the delivery sequence of retailers in the tour. $Seq(h)$ represents the h-th delivered retailer in the tour, and $Seq(0)$ represents the factory index.

Algorithm 2 DPT function.

```
$FRT$: Stores the time from factory to retailer in the tour.
$SP$: Stores the tour departure times corresponding to soft time windows (segmentation points).
For $h \in \{1, \cdots, H\}$ do
    $FRT = 0$;
    For $i \in \{0, \cdots, h-1\}$ do
        $FRT = FRT + T_{Seq(i),Seq(i+1)}$;
    End for
    If $E_{Seq(h)} - FRT$ > 0 then
        Put $E_{Seq(h)} - FRT$ in $SP$;
    Else
        Put $0$ in $SP$;
    End if
    If $L_{Seq(h)} - FRT > 0$ then
        Put $L_{Seq(h)} - FRT$ in $SP$;
    Else
        Put $0$ in $SP$;
    End if
End for
Remove duplicate time nodes from $SP$ and then the $DPT_{tour}$ is calculated for each time node in $SP$.
The time node (or interval) corresponding to the minimized $DPT_{tour}$ is the ideal optimal departure time.
```

4.2.3. *The simultaneous coordination of batch manufacturing and tour departure schedules: BSS-NWS*

A batch is delivered immediately after it is manufactured which is called a no-wait schedule. It does not generate inventory, so this is an ideal way of handling it. Due to the practical limitations of production capacity and the number of vehicles, BSS-NWS is proposed to satisfy the no-wait schedule as much as possible. It consists of three sub-strategies, and its specific process is shown in the corresponding part of Fig. 4. Simultaneous coordination of production and delivery is achieved through the effective integration of three sub-strategies, and the propositions given below establish this concept.

(1) Sub-strategy 1: The maximum adjustment range strategy.

The following symbols are introduced: $\left[IDT^e_{B(j)}, IDT^l_{B(j)}\right]$ represents the ideal optimal departure time interval of batch $B(j)$. $tr_{v,B(i)}$ represents the time when the vehicle returns to the factory after the delivery of batch $B(i)$ is completed. $td^e_{B(j)}$ represents the earliest departure time of batch $B(j)$.

The sub-strategy1 execution steps are as follows: First, starting from time node 0, the batches are continuously arranged in the manufacturing sequence thereby obtaining the earliest completion time node of batches. Then, in order to obtain the maximum adjustment range to achieve the minimum inventory backlog, according to $IDT^l_{B(j)}$ and $td^e_{B(j)}$,$td_{B(j)} = \max\left(td^e_{B(j)}, IDT^l_{B(j)}\right)$ is updated in the sequence of batch manufacturing. Finally, according to the no-wait schedule, we perform backward scheduling of batches in reverse manufacturing sequence, and $tc_{p,B(j)} = \min\left(ts_{p,B(j+1)}, td_{B(j)}\right)$ is updated.

Proposition 3. *If batch B(i) is immediately followed by batch B(j) during a vehicle, then according to the no-wait schedule,*$td^e_{B(j)} = \max\left(tr_{v,B(i)}, \max\left(tc_{1,B(j)}, \cdots, tc_{P,B(j)}\right)\right)$.

Proof. Batch delivery requires two conditions to be met: 1) All products from the batch are produced. 2) The selected vehicle of the batch is currently available in the factory. If the tour corresponding to batch $B(i)$ is the first tour of vehicle v,$tr_{v,B(i)} = 0$, otherwise $tr_{v,B(i)}$ is non-zero. In order to obtain a no-wait schedule, $td^e_{B(j)} = \max\left(tr_{v,B(i)}, \max\left(tc_{1,B(j)}, \cdots, tc_{P,B(j)}\right)\right)$. QED.

Proposition 4. *By considering the latest node of the ideal departure time* $IDT^l_{B(j)}$*, the strategy of maximum adjustment range can achieve the minimum inventory backlog.*

Proof. For tours, the optimal departure time can be obtained at any node within $\left[IDT^e_{B(j)}, IDT^l_{B(j)}\right]$, but each node is different for the production line. Affected by the time constraints on the production line, if we use nodes earlier than $IDT^l_{B(j)}$ for calculation, the corresponding departure time $td_{B(j)}$ of the batch may be advanced. When $tc_{p,B(j)}$ is updated in the reverse sequence of manufacturing, the scope of backward adjustment for each batch becomes smaller. When $tc_{p,B(j)} < IDT^e_{B(j)}$, the batch will incur an inventory backlog. Therefore, we use the latest time node $IDT^l_{B(j)}$ to achieve the minimum inventory backlog while trying to meet the no-wait schedule. QED.

To illustrate the above process, we give an example in Table 2. As shown in Fig. 6, sub-strategy1 is executed according to $IDT^l_{B(j)}$. If adjusted according to $IDT^e_{B(j)}$, the advance of $tc_{p,B(j)}$ causes the inventory backlog of $B(1, 1)$ and $B(2, 1)$. Obviously, the later the delivery time node is set, the larger the adjustable range, and the less possibility of causing inventory backlog.

(2) Sub-strategy 2: The earliest delivery time strategy.

Since sub-strategy 1 use $IDT^l_{B(j)}$ to obtain the maximum adjustment range, the time span of production and delivery is long. In order to ensure that batches can be produced and delivered earlier based on the no-wait schedule and no inventory generation, we propose the following sub-strategy 2.

Table 2
BSS-NWS example data.

Batch	Production time		Routing time	$\left[IDT^e_{B(j)}, IDT^l_{B(j)}\right]$
	Production line 1	Production line 2		
B1	30	21	36	[50, 71]
B2	54	21	45	[100, 130]
B3	50	30	20	[147, 177]

We perform forward scheduling of batches according to the manufacturing sequence. When there is an idle time between two adjacent batches on a production line, it means that the production start time of the next batch can be advanced. The production completion time of the batch after the advancement is expressed by $tc^a_{p,B(j)} = tc_{p,B(j)} - \left(ts_{p,B(j)} - tc_{p,B(j-1)}\right)$. Hence, the departure time of the batch is advanced $td^a_{B(j)} = \max\left(tr_{v,B(i)}, \max\left(tc^a_{1,B(j)}, \cdots, tc^a_{P,B(j)}\right)\right)$. Comparing with $IDT^e_{B(j)}$, we can update $td_{B(j)} = \max\left(td^a_{B(j)}, IDT^e_{B(j)}\right)$. Let $tc_{p,B(j)} = \min\left(ts_{p,B(j+1)}, td_{B(j)}\right)$ according to the no-wait schedule.

Proposition 5. *The production and delivery time nodes of each batch can only be adjusted forward or remain unchanged. Therefore, the time span of production and delivery will be reduced or remain unchanged.*

Proof. According to the manufacturing sequence, $ts_{p,B(j)} \geqslant tc_{p,B(j-1)}$. And $tc^a_{p,B(j)} = tc_{p,B(j)} - \left(ts_{p,B(j)} - tc_{p,B(j-1)}\right)$, so $tc^a_{p,B(j)} \leqslant tc_{p,B(j)}$, $td^a_{B(j)} = \max\left(tr_{v,B(i)}, \max\left(tc^a_{1,B(j)}, \cdots, tc^a_{P,B(j)}\right)\right) \leqslant \max\left(tr_{v,B(i)}, \max\left(tc_{1,B(j)}, \cdots, tc_{P,B(j)}\right)\right) = td^e_{B(j)}$. Before adjustment, $td_{B(j)}$ is equal to $\max\left(td^e_{B(j)}, IDT^l_{B(i)}\right)$, and after adjustment, $td_{B(j)}$ is equal to $\max\left(td^a_{B(j)}, IDT^e_{B(j)}\right)$. Therefore, $tc_{p,B(j)}$ and $td_{B(j)}$ can only be adjusted forward or remain unchanged. QED.

(3) Sub-strategy 3: The trade-off strategy of inventory cost and production start-up cost.

Batches are often produced discontinuously after the first execution of sub-strategy 2. Production start-up costs can be reduced by eliminating the idle time between adjacent batches on the production line, but batch advance production will incur new inventory costs. We need to compare the two costs to determine whether the batch should be advanced. In addition, $tc_{p,B(j)}$ no longer corresponds to $td_{B(j)}$ when there is batch $B(j)$ for advanced production. It is necessary to determine whether sub-strategy 2 requires to be executed again for the better meeting of the no-wait schedule. Hence, the optimal integration solution is obtained by cycling sub-strategies 2 and 3 until no batch can be produced in advance.

Sub-strategies 2 and 3 are executed cyclically as follows: Sub-strategy 3 is executed to adjust all batches $B(I, P)$ on the production line in the manufacturing sequence. $B(I, P)$ is adjusted forward when the inventory cost incurred by the elimination of idle time is less than the production start-up cost. To better meet the no-wait schedule, sub-strategy 2 is executed for $B(I, P)$ which is not adjusted forward. Then, sub-strategies 2 and 3 are executed cyclically until there is no batch forward adjustment. Proposition 6 shows that this cyclic process can be completed in polynomial time.

To illustrate the above process, we follow the previous example, the data for which is shown in Table 2. As shown in Fig. 7, through the execution of sub-strategy 2, the integrated solution enters stage 2. Then, sub-strategy 3 is executed sequentially for each batch on the production line. It is assumed that the production start-up cost of production line 1 is greater than the inventory cost incurred by batch $B(1, 1)$ in advance, and $B(1, 1)$ is adjusted forward into Stage 3. It is worth noting that batches *B2* and *B3* can be further optimized by sub-strategy 2 in stage 3.

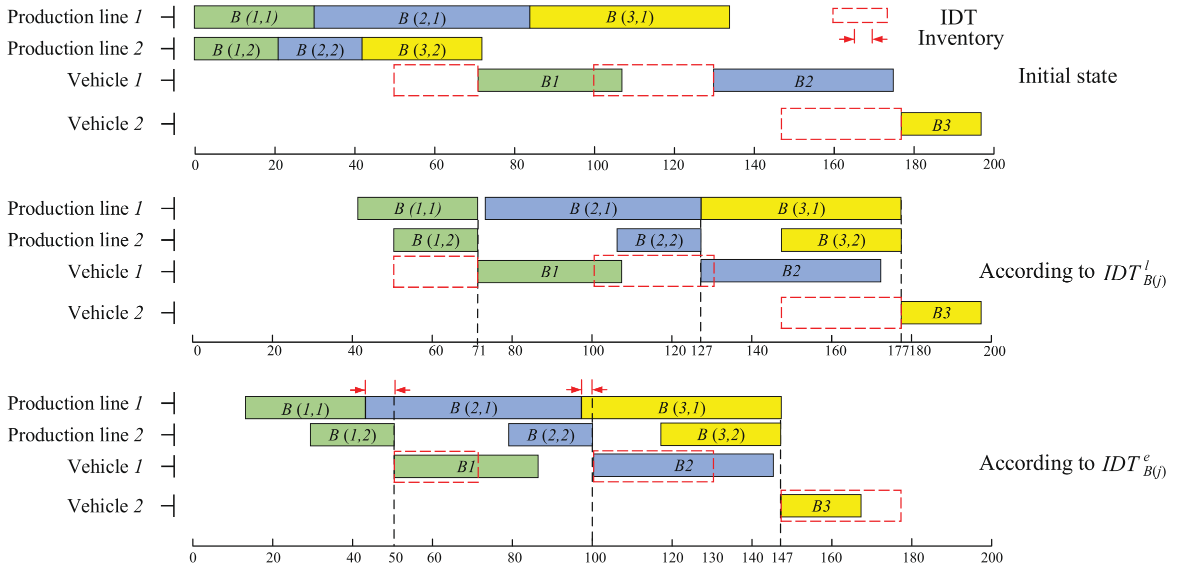


Fig. 6. Comparison of different time nodes.

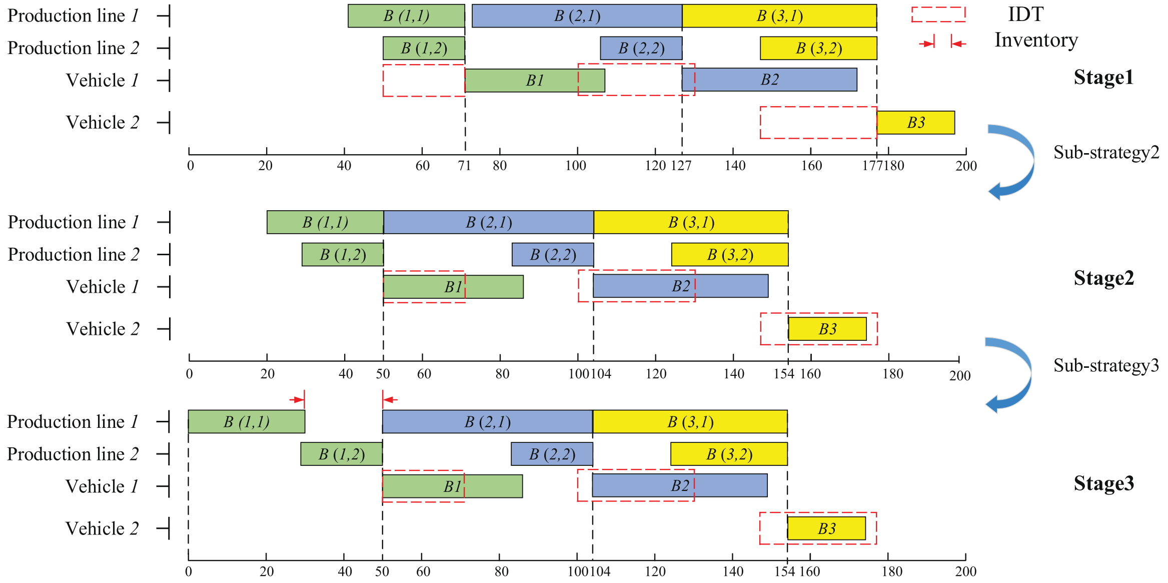


Fig. 7. Execution of sub-strategies 2 and 3.

The production completion and departure times for batches *B2* and *B3* will be advanced to 100 and 150, respectively. Re-execute sub-strategy 3 until the integrated solution is not changed.

Proposition 6. *After sub-strategies 2 and 3 are called at most I*P times in a loop, we can obtain the final integration solution. Its time complexity is O (I*P).*

Proof. There are *P* production lines and *I* batches. Sub-strategy 3 is to eliminate idle time according to the manufacturing sequence. Obviously, the time node of the batch is adjusted forward. According to Proposition 5, sub-strategy 2 is also adjusted forward. And each time the adjustment is made, the time node of the first forward-adjusted *B* (*I*, *P*) will not change. Since at least one *B* (*I*, *P*) is adjusted forward at a time, the cycle is executed at most *I***P* times. Its time complexity is *O*(*I***P*), and the integrated solution can be obtained in polynomial time. QED.

4.3. Realization of holistic iterative optimization mode based on ALNS

4.3.1. Neighborhood structures of ALNS

Traditional ALNS have multiple destroy and repair operators. Relatively independent destroy operators and repair operators can be linearly combined to form a variety of neighborhood structures. Weights are assigned to each destroy and repair operator (Aksen et al., 2014; Pisinger and Ropke, 2007). A criticism of the traditional ALNS is that the random combination of destroy and repair operators makes the search random and blind, which may slow down the optimization algorithm. To search more targeted, we design four destroy-repair neighborhood structures considering the problem characteristics and coding form, where the destroy operator and the repair operator have a one-to-one correspondence. And weights are assigned to the neighborhood structures. Two neighborhood structures are designed based on the delivery vehicle and heterogeneous capacity characteristics to enhance the exploitation ability of the algorithm. Two neighborhood structures are designed based on the coding form for expanding the search region and adjusting the Pareto front, enhancing the exploration capability of the algorithm.

In order to illustrate the execution process of the neighborhood structure, the following elements are defined: The individual set of the Pareto front $\xi^P = \{\xi_1^P, \cdots \xi_r^P, \cdots \xi_R^P\}$ is the execution object of the neighborhood structure, where ξ_r^P represents the rth individual on the Pareto front. ξ_1^{new}, ξ_2^{new} are used to store the new neighborhood individuals generated by performing the neighborhood search for ξ_r^P. The neighborhood search scale (Ns) represents the number of times ξ_r^P executes the neighborhood structure, and NIS represents the neighborhood individual set.

(1) Problem-based design of neighborhood structures.

Neighborhood structure 1: Aiming at optimizing the use of delivery vehicles to avoid continuous delivery of the same vehicle. Neighborhood structure 1 is designed in Algorithm 3. For each individual in ξ^P, if adjacent batches are delivered by the same vehicle, the non-first batch is very likely to incur inventory backlog and delivery time tardy. Therefore, replacing the available delivery vehicle for one of the batches has a high probability of optimizing the current individual when there are adjacent batches delivered by the same vehicle. Moreover, two replacement methods are applied: The whole batch replacement and the replacement of each order within the batch.

Algorithm 3 Continuous delivery vehicle replacement.

```
$NIS = \varnothing$;
For $r \in \{1, \cdots, R\}$ do
$\xi_1^{new} = \varnothing, \xi_2^{new} = \varnothing$;
  If there are adjacent batches delivered by the same vehicle in $\xi_r^P$ then
      $\xi_1^{new}$ ← Randomly select one of the batches, and replace the delivery vehicle for the whole batch.
      $\xi_2^{new}$ ← Randomly select one of the batches, and replace the delivery vehicle for each order.
      $NIS = [NIS, \xi_1^{new}, \xi_2^{new}]$;
  End if
End for
Obtain $NIS$.
```

Neighborhood structure 2: Since a limited number of heterogeneous vehicles are used for delivery, it is critical to select the appropriate vehicle based on the batch size. In order to improve the loading rate of delivery vehicles for reducing distribution costs, neighborhood structure 2 is designed in Algorithm 4. The lower the actual load rate of the delivery vehicle in the tour plan, the greater the probability that the retailer's order in the tour will be selected to replace the delivery vehicle. The two replacement methods are still applied: The whole batch replacement and the replacement of each order within the batch.

Algorithm 4 Improve vehicle load operation.

```
$C_v$: Maximum load number of pallets for vehicle $v$.
$BO_v^{B(i)}$: The number of pallets occupied by batch $B(i)$delivered by vehicle $v$.
$BOR_{B(i)}$: Actual occupancy rate of batch $B(i)$.
$NIS = \varnothing$;
For $r \in \{1, \cdots, R\}$ do
  Get the order composition of all batches $B(i)$ in $\xi_r^P$.
  $BOR_{B(i)} = BO_v^{B(i)} / C_v$;
  $k = 1$;
  While $k \leqslant Ns$ do
      $\xi_1^{new} = \varnothing, \xi_2^{new} = \varnothing$;
      Select a batch for vehicle replacement by roulette wheel strategy based on $BOR_{B(i)}$.
      $\xi_1^{new}$ ← Randomly replace the delivery vehicles of the selected batch according to $C_v$.
```

(continued on next column)

(continued)

```
      $\xi_2^{new}$ ← Randomly replace the delivery vehicle for each order in the selected batch.
      $NIS = [NIS, \xi_1^{new}, \xi_2^{new}]$;
      $k = k + 1$;
  End while
End for
Obtain $NIS$.
```

(2) Coding-based design of neighborhood structures.

Neighborhood structure 3: To expand the search region and help the algorithm to jump out of the local optimum, neighborhood structure 3 is designed in Algorithm 5. We randomly select part of the coding positions to perform separation and cohesion operations. And the vehicle serial numbers with the lowest and highest frequency of occurrence are obtained for all individuals within ξ^P at the selected coding positions. The separation operation is to replace the selected coding position with the vehicle serial number with the lowest frequency. The cohesion operation is to replace the selected coding position with the vehicle serial number with the highest frequency. If the frequency of occurrences is the same, it is selected randomly.

Algorithm 5 Separation-Cohesion operation.

```
$NIS = \varnothing$;
For $r \in \{1, \cdots, R\}$ do
  $k = 1$;
  While $k \leqslant Ns$ do
      $\xi_1^{new} = \varnothing, \xi_2^{new} = \varnothing$;
      Randomly select part of the code positions in $\xi_r^P$ for replacing the vehicle serial number.
      $\xi_1^{new}$ ← The separation operation is performed to generate a new neighborhood individual.
      $\xi_2^{new}$ ← The cohesion operation is performed to generate a new neighborhood individual.
      $NIS = [NIS, \xi_1^{new}, \xi_2^{new}]$;
      $k = k + 1$;
  End while
End for
Obtain $NIS$.
```

Neighborhood structure 4: In order to make the Pareto front more uniform and broader, the individual search is conducted towards the sparse region and away from the crowded region by biased adjustment of the individual on the Pareto front. The crowding distance assignment method (Deb et al., 2002) is adopted to measure the crowding degree of individuals. We regard the least crowded individual on the Pareto front as the attractive individual ξ_{AT}^P, and the relatively crowded individual as the repulsive individual ξ_{RE}^P. For each individual ξ_r^P, the search is performed in the direction of approaching ξ_{AT}^P and away from ξ_{RE}^P. Algorithm 6 gives the specific execution process.

Algorithm 6 Attraction-Repulsion operation.

```
$NIS = \varnothing$;
Calculate the crowding distance ($Crow_r$) for each individual on the Pareto front.
The individual with minimized $Crow_r$ is regarded as $\xi_{AT}^P$.
One of the individuals with $Crow_r \geqslant 0.5$ is randomly selected as $\xi_{RE}^P$.
For $r \in \{1, \cdots, R\}$ do
  $k = 1$;
  While $k \leqslant Ns$ do
      $\xi_1^{new} = \varnothing, \xi_2^{new} = \varnothing$;
      Obtain the code positions of different vehicle serial numbers in $\xi_r^P$ and $\xi_{AT}^P$.
      $\xi_1^{new}$ ← Randomly select a part of the obtained code positions and change their vehicle serial numbers to be the same as the corresponding code positions in $\xi_{AT}^P$.
```

(continued on next page)

(*continued*)

```
        Obtain the code positions of the same vehicle serial numbers in $\xi_j$ and $\xi_{RE}^{P}$.
        $\xi_2^{new}$ ← Randomly select a part of the obtained code positions and randomly change their vehicle serial numbers.
        $NIS = [NIS, \xi_1^{new}, \xi_2^{new}]$;
    $k = k + 1$;
End while
End for
Obtain NIS.
```

4.3.2. *Main components of ALNS*

(1) Large neighborhood: During the execution of the neighborhood search, the search direction varies because the four neighborhood structures (rules) have different purposes. And an indefinite number of retailers' delivery vehicle serial numbers are changed to generate new neighborhood individual solutions. Therefore, the range of the neighborhood search is large and diverse.

(2) Adaptive search engine: At each iteration, each neighborhood structure is assigned a weight, and the choice of neighborhood structure is controlled by the roulette-wheel mechanism. Let WNS_q be the weight assigned to neighborhood structure q based on its past performance. Then the neighborhood structure is chosen with probability $WNS_q/\sum_q^4 WNS_q$.

(3) Adaptive weight adjustment: At the beginning, each neighborhood structure is assigned the same weight. In each iteration, the Pareto front is updated after the execution of the neighborhood structure. The adaptive weight adjustment is conducted according to the update level of individuals on the Pareto front. The number of added individuals on the Pareto front is set to Num_a, and the ratio of the removed individual numbers to the original total individual numbers is set to *Rat*. The increased score of the neighborhood structure is derived according to Eq. (24), and then WNS_q is updated. Clearly, Num_a must be more than 0 when *Rat* >0. And the larger the Num_a and *Rat*, the better the performance of the executed neighborhood structure. Thus, the design of $0 = \omega 1 < \omega 2 < \omega 3 < \omega 4 < \omega 5 < \omega 6$ is reasonable.

$$\omega = \begin{cases} \omega 1, & Rat = 0 \text{ and } Num_a = 0 \\ \omega 2, & Rat = 0 \text{ and } Num_a > 0 \\ \omega 3, & 0 < Rat \leqslant 0.25 \\ \omega 4, & 0.25 < Rat \leqslant 0.5 \\ \omega 5, & 0.5 < Rat \leqslant 0.75 \\ \omega 6, & Rat > 0.75 \end{cases} \tag{24}$$

5. Computational experiments

In this section, we show detailed experiments to evaluate the performance of the proposed IMOEA. We compared IMOEA with NSGA-II (Ganji et al., 2020), the discrete differential evolution algorithm based on multi-objective optimization (MODE) (Qu et al., 2020), HGAPSO1 (Soleimani and Kannan, 2015), HGAPSO2 (Biuki et al., 2020) and an improved multi-objective differential evolutionary algorithm (IMODE) is formed by adopting the proposed ALNS to improve MODE. All experiments are implemented by using MATLAB2020a and run on an Intel Core i5-1035G1-CPU 1.00 GHz and 16.0 GB of RAM.

5.1. *Testing instances*

We generate the test instances according to the method proposed by Yagmur and Kesen (2023) for making a fair comparison. And they show that the true Pareto solution cannot be found by exact traversal solving in two hours when the number of retailers exceeds 10. Instances (*P-V-n-D*) are formed by combining the number of production lines *P*, the delivery vehicle *V*, the number of retailers *n*, and the retailer product demand *D*. In the investigated enterprise, the number of regularly running production lines is 3–5 ($P \in \{3, 4, 5\}$). And the delivery of orders is performed by 4–6 vehicles ($V \in \{4, 5, 6\}$), whose capacity varies from 24–31 pallets. In addition, the number of retailers *n* is the main factor distinguishing the scale of the instance. From the perspective of model solving, the larger *n* is, the longer the encoding individual (one-dimensional vector) is, and the larger the solution space is. Therefore, the small-, medium- and large-scale instances contain 20, 25, and 30 retailers respectively. A total of 30 instances, 9 small-scale, 9 medium-scale, and 12 large-scale, are generated.

Table 3
The generation of problem parameters.

Parameter	Generation procedure
$D_{k,p}$	$\sim N(\alpha, \alpha^2)$
Pt_p	$\sim U(1/\alpha, 4/\alpha)$
$[E_k, L_k]$	$E_k \sim U(15, 100)$ $L_k - E_k \in \{0.5; 1; 1.5; 2; 2.5; 3\}$
O_p	$\in \{1/\alpha; 5/4\alpha; 5/3\alpha; 2/\alpha\}$
C_v	$\in \{24; 26; 28; 29; 31; 33\}$

The specific method for determining the problem parameters is as follows: First, we calculate the travel distance between each site by locating the latitude and longitude of the real factory and retailer sites, thereby getting the travel time matrix $T_{k,l}$. In addition, some crucial parameters are set as shown in Table 3. We define a general α whose value is set to 1000. The demand size of each retailer for different products follows a normal distribution $N(\alpha, \alpha^2)$. The product processing time is uniformly distributed between $1/\alpha$ and $4/\alpha$. The lower bound E_k of each retailer's time window is generated from uniform distribution *U* (*15*, *100*), and L_k is generated from E_k. The selection range for the width of time windows ($L_k - E_k$), the pallet capacity occupied by each product (O_p), and the load number of pallets for the vehicle (C_v) are given in Table 3. The dataset can be downloaded from https://www.huangm.cn/zip/IPSVRP-BM-STW-Dateset.zip.

5.2. *Performance metrics*

In order to comprehensively evaluate the performance of the algorithm, four metrics are used to evaluate the cardinality, convergence, distribution, and spread of algorithms, i.e., Error ratio (ER) (Audet et al., 2021), Distribution metric (DM) (Zheng et al., 2016), Inverted generational distance(IGD) (Coello Coello and Cruz Cortés, 2005), Hypervolume (HV) (Zitzler et al., 2000). Y_c represents the current non-dominated solution set and Y_r represents the reference non-dominated solution set. The Eqs. (25)-(28) are given as follows:

$$ER(Y_c) = \frac{|\{y \in Y_c | y \notin Y_r\}|}{|Y_c|} \tag{25}$$

$$DM(Y_c) = \frac{1}{|Y_c|} \sum_{i=1}^{m} \left(\frac{\sigma_i}{\mu_i}\right) \left(\frac{|y_i^U - y_i^N|}{\max\limits_{y \in Y_c} y_i - \min\limits_{y \in Y_c} y_i}\right)$$

$$\sigma_i = \frac{1}{|Y_c| - 2} \sum_{j=1}^{|Y_c|-1} (d_i^j - \overline{d}_i)^2 \mu_i = \frac{1}{|Y_c| - 1} \sum_{j=1}^{|Y_c|-1} d_i^j \tag{26}$$

$$IGD(Y_c, Y_r) = \frac{1}{|Y_c|} \left(\sum_{y^2 \in Y_r} \left(\min_{y^1 \in Y_c} \|y^1 - y^2\| \right)^m \right)^{\frac{1}{m}} \tag{27}$$

$$HV(Y_c; r) = \lambda \left(\bigcup_{y \in Y_c} [y, r] \right) \tag{28}$$

Metric ER is used to evaluate the cardinality of the algorithm, the lower the metric value, the better it is considered. Metric DM is used to evaluate the distribution and spread, and to judge whether the individuals on the Pareto front are more uniform and broader. A smaller

Table 4
Computational results of all algorithms.

Scale	Instance	Obj	IMOEA			NSGA-II			MODE			IMODE			HGAPSO1			HGAPSO2		
			Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)
Small	3–4-	*f1*	**29.6729**	**27.5221**	83.8	29.1091	27.1434	56.7	28.5159	26.6016	51.5	**29.6729**	27.4502	78.3	28.9297	26.7667	112.1	29.6534	26.8900	83.6
	20–1	*f2*	**0.3339**	0.4883		0.3343	0.4826		0.3684	0.5065		**0.3339**	**0.4813**		0.3428	0.5063		**0.3339**	0.5027	
	3–4-	*f1*	**32.5643**	**31.2166**	53.4	32.2198	31.0335	32.4	31.4319	30.3420	38.1	32.4362	31.1053	61.2	32.0945	30.6434	95.2	32.4633	31.1439	56.7
	20–2	*f2*	**1.8943**	2.1643		**1.8943**	2.1595		1.9903	2.1596		**1.8943**	**2.1389**		1.9268	2.1924		**1.8943**	2.1658	
	3–4-	*f1*	**25.1696**	**23.5569**	122.6	24.9985	23.0484	57.6	24.1724	22.5124	59.8	25.1156	23.4790	107.6	24.7878	22.8749	91.3	25.1415	23.0306	81.3
	20–3	*f2*	**0.5836**	0.7700		**0.5836**	**0.7471**		0.6124	0.7654		**0.5836**	0.7788		0.5871	0.8061		0.5853	0.7571	
	4–4-	*f1*	**34.9078**	**33.6819**	67.8	34.5783	33.2209	38.6	33.9352	32.7618	45.7	34.8316	33.5814	74.5	34.4258	33.0489	94.4	34.5838	33.3273	72.5
	20–1	*f2*	**1.9725**	2.2854		**1.9725**	2.2383		2.0229	2.2407		**1.9725**	2.2662		1.9898	2.2811		**1.9725**	**2.2211**	
	4–4-	*f1*	**35.2418**	33.9473	64.5	34.9576	33.6765	40.0	34.3227	33.1747	46.0	35.2057	**33.9644**	61.9	34.8034	33.4932	102.3	34.9869	33.7916	72.7
	20–2	*f2*	**2.0420**	2.3224		**2.0420**	**2.3053**		2.0845	2.3327		**2.0420**	2.3264		2.0568	2.3525		**2.0420**	2.3101	
	4–4-	*f1*	**35.5526**	**34.4982**	70.4	35.3585	34.3406	38.3	34.6286	33.6590	45.9	35.5155	34.4221	73.5	35.1420	33.8565	83.7	35.5362	34.3073	70.5
	20–3	*f2*	**2.1063**	2.3857		2.1119	2.4011		2.1457	2.3829		**2.1063**	2.3943		2.1139	2.4089		**2.1063**	**2.3699**	
	5–5-	*f1*	**34.5935**	**33.2449**	90.7	34.1981	32.7233	39.4	33.5833	32.2086	38.1	34.4445	33.0348	101.9	34.0495	32.5119	118.6	34.5104	32.8047	86.5
	20–1	*f2*	**2.0283**	2.2753		**2.0283**	2.2393		2.0642	2.2713		**2.0283**	2.2498		2.0622	2.3196		2.0288	**2.2380**	
	5–5-	*f1*	33.9991	**32.8906**	65.8	33.6327	32.4290	24.0	32.9764	31.7918	35.0	**34.0312**	32.7801	62.8	33.3907	32.1020	99.3	33.7944	32.3848	60.1
	20–2	*f2*	**1.9178**	2.1511		1.9267	2.1402		1.9685	2.1551		1.9205	2.1536		1.9467	2.2122		1.9191	**2.1136**	
	5–5-	*f1*	31.3037	**29.4783**	80.0	30.9489	28.7944	35.5	30.0595	28.4277	30.0	**31.3439**	29.4224	96.6	30.5620	28.4085	108.3	30.8598	28.6934	84.0
	20–3	*f2*	**0.0059**	0.0847		0.0131	0.0876		0.0229	0.1081		**0.0059**	0.0870		0.0140	0.1035		0.0098	**0.0814**	
Medium	3–5-	*f1*	30.1163	**28.1717**	104.5	29.4742	27.3813	51.2	29.0096	26.9659	47.2	**30.1246**	28.0138	124.2	28.9846	27.0944	122.3	29.6254	27.5383	95.9
	25–1	*f2*	**0.7945**	**0.9496**		0.8405	0.9852		0.8601	0.9996		0.8328	0.9652		0.8274	0.9978		0.8278	0.9891	
	3–5-	*f1*	39.7686	**38.3174**	75.3	39.4166	37.9894	36.8	38.6299	37.5677	26.9	**39.8230**	38.2905	70.4	39.1533	37.6218	103.2	39.7489	38.0734	73.1
	25–2	*f2*	3.1101	**3.3848**		3.1957	3.4807		3.3116	3.5187		**3.0818**	3.3922		3.2278	3.5282		3.1578	3.4162	
	3–5-	*f1*	**40.3849**	38.7948	93.0	39.8260	38.4426	41.7	39.0711	37.9111	36.2	40.3705	**38.8572**	94.6	39.5312	38.0468	108.2	40.1770	38.5422	85.2
	25–3	*f2*	**3.0864**	**3.3513**		3.1365	3.3926		3.2588	3.4709		3.0961	3.3767		3.1915	3.4625		3.1063	3.4182	
	4–5-	*f1*	**44.0469**	**42.3952**	111.6	43.4300	42.0737	33.8	42.6712	41.5031	41.9	**44.0469**	42.2953	125.0	43.0777	41.8443	80.0	43.7310	41.9022	68.8
	25–1	*f2*	3.5382	3.8276		3.6213	3.8480		3.7781	3.9728		**3.5332**	**3.8158**		3.6594	3.9870		3.5889	3.8545	
	4–5-	*f1*	**42.9189**	**41.3423**	69.1	42.6204	41.0363	36.7	41.7669	40.4078	28.2	42.8997	41.2491	75.8	42.1210	40.7587	116.6	42.4529	40.9763	52.6
	25–2	*f2*	**3.3168**	3.6057		3.3855	3.6486		3.5680	3.7716		3.3257	**3.6003**		3.4330	3.7265		3.3952	3.6710	
	4–5-	*f1*	**42.8171**	40.8576	86.7	42.3127	40.8418	36.8	41.5764	40.1901	27.7	42.7211	**40.9102**	92.7	42.0368	40.6100	117.9	42.5117	40.6675	53.4
	25–3	*f2*	**3.2558**	3.5203		3.3125	3.5503		3.4466	3.6554		3.2648	**3.5077**		3.3749	3.6308		3.2989	3.5793	
	5–5-	*f1*	**42.7096**	**41.0969**	88.3	42.0447	40.7334	41.0	41.4395	40.2043	28.5	42.6211	41.0837	88.6	41.9860	40.4535	118.9	42.5561	40.6542	64.0
	25–1	*f2*	**3.4645**	**3.7443**		3.5877	3.8177		3.7092	3.9354		3.4984	3.7791		3.5461	3.8617		3.5062	3.8398	
	5–5-	*f1*	43.5454	**41.8226**	86.1	43.3003	41.7989	38.5	42.2232	41.0252	30.6	**43.5881**	41.7592	96.0	42.9561	41.5468	123.3	43.5629	41.5828	66.0
	25–2	*f2*	**3.7470**	**4.0052**		3.8861	4.0999		3.9676	4.1916		3.7969	4.0194		3.8842	4.1942		3.8056	4.0845	
	5–5-	*f1*	43.1884	41.5273	67.2	42.6723	41.3160	35.4	41.9015	40.8394	27.4	**43.2125**	**41.5343**	82.0	42.4696	41.1172	115.7	43.1191	41.1213	55.4
	25–3	*f2*	**3.7210**	**3.9818**		3.7761	4.0768		3.9622	4.1937		3.7311	4.0071		3.8735	4.1373		3.7702	4.0481	
Large	3–5-	*f1*	**43.5061**	**40.9025**	125.8	42.9090	40.8042	44.0	41.8997	40.2908	36.4	43.4590	40.8288	117.8	42.5200	40.2801	151.9	43.0482	40.4234	77.3
	30–1	*f2*	**2.4250**	**2.6519**		2.4705	2.6879		2.6007	2.8645		2.4416	2.6574		2.4614	2.7791		2.4324	2.7498	
	3–5-	*f1*	**43.7864**	41.1395	165.3	43.1416	**41.1638**	51.7	41.9769	40.3063	44.2	43.7545	41.0773	176.9	42.5164	40.3158	148.4	43.1267	40.6676	75.2
	30–2	*f2*	**2.3478**	**2.5752**		2.3687	2.6556		2.4363	2.7486		2.3616	2.5812		2.3945	2.6829		2.3721	2.6345	
	3–5-	*f1*	42.9977	40.4843	125.0	42.4661	40.3720	42.3	41.0623	39.6057	31.0	**43.0182**	**40.5757**	147.2	41.9744	39.6933	146.3	42.2426	39.8379	69.2
	30–3	*f2*	**2.2986**	**2.5090**		2.3361	2.6051		2.4539	2.7045		2.3045	2.5450		2.3422	2.6353		2.3328	2.5676	
	4–6-	*f1*	**47.8534**	**45.2606**	144.3	46.7782	44.8392	49.5	45.6284	44.1060	44.7	47.5255	45.2173	147.0	46.4822	44.1879	164.8	47.0085	44.5671	82.7
	30–1	*f2*	**2.6741**	2.8720		2.7088	**2.8655**		2.7910	3.0088		**2.6741**	2.8779		2.7207	2.9807		2.6962	2.8807	
	4–6-	*f1*	47.5617	**45.0799**	164.2	46.7534	44.5923	51.6	45.4496	43.7100	42.7	**47.5664**	44.8650	193.6	45.9606	43.9255	186.4	46.7279	44.2242	89.8
	30–2	*f2*	**2.6392**	**2.8433**		2.6721	2.8721		2.7697	2.9841		2.6473	2.8690		2.6981	2.9484		2.6583	2.8671	
	4–6-	*f1*	**46.1832**	**43.8195**	173.9	45.5572	42.8062	58.8	44.5334	42.7415	35.5	46.0415	43.6667	192.1	44.8325	42.5781	177.0	45.5599	42.9474	81.7
	30–3	*f2*	**2.4691**	**2.6507**		2.4996	2.6750		2.5783	2.8478		2.4735	2.6768		2.5254	2.7605		2.5176	2.6939	

(*continued on next page*)

Table 4 (*continued*)

Scale	Instance	Obj	IMOEA			NSGA-II			MODE			IMODE			HGAPSO1			HGAPSO2		
			Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)	Best	Avg	Time (sec)
	5–5-30–1	*f1*	**47.0444**	44.1049	147.1	46.5381	43.9812	52.5	45.4380	43.4158	43.0	46.8763	**44.3385**	180.8	45.9171	43.5402	190.1	46.3065	43.2833	91.1
		f2	**2.6945**	**2.9072**		2.7260	3.0181		2.8223	3.0964		2.7007	2.9568		2.7248	3.0539		2.7098	2.9683	
	5–5-30–2	*f1*	**46.7432**	44.0992	178.6	46.4529	43.7507	52.4	45.1902	42.9667	43.5	46.7192	**44.1511**	170.8	45.6571	43.2887	193.8	45.9791	43.1232	101.0
		f2	**2.5520**	**2.7933**		2.5838	2.8419		2.6833	2.9576		2.5694	2.8182		2.6160	2.9221		2.6019	2.8355	
	5–5-30–3	*f1*	**46.7083**	**43.9373**	208.8	45.8213	43.6349	54.3	44.9398	43.1219	43.2	46.5839	43.7823	227.8	45.6665	43.2142	209.1	46.0585	43.1493	106.7
		f2	**2.6400**	**2.8737**		2.6761	2.9328		2.8040	3.0658		2.6537	2.8917		2.7223	3.0166		2.6784	2.9315	
	5–6-30–1	*f1*	**47.5447**	44.8472	127.7	46.8446	44.4569	43.2	45.5618	43.7696	34.4	47.4755	**45.1421**	131.9	46.3904	43.7344	169.7	46.5109	43.8589	80.5
		f2	**2.6283**	**2.8218**		2.6503	2.8923		2.7596	3.0132		2.6359	2.8974		2.6874	2.9413		2.6638	2.8732	
	5–6-30–2	*f1*	**48.0325**	**45.2735**	145.7	47.4602	44.9852	45.6	45.8039	44.2493	35.9	47.7309	45.1991	163.0	46.6254	44.3017	158.9	47.1036	44.3795	72.4
		f2	**2.8024**	**3.0008**		2.8110	3.0628		2.9326	3.2114		2.8100	3.0339		2.8608	3.1599		2.8217	3.0715	
	5–6-30–3	*f1*	**47.9307**	45.1380	154.1	47.1306	44.7287	45.6	45.9534	44.1297	35.9	47.7149	**45.2436**	160.1	46.6101	44.3342	194.9	47.0278	44.4598	93.1
		f2	**2.7291**	**2.9402**		2.7536	3.0030		2.8766	3.1246		2.7391	2.9640		2.7962	3.1011		2.7340	2.9997	

The bold value means the best performance, and all best and average values are in scientific notation ($\times 10^4$).

DM indicates better distributed solutions. Both IGD and HV are used to evaluate convergence and distribution. IGD requires the true Pareto frontier, since it cannot be obtained, it is often represented by the reference Pareto front obtained by all algorithms. HV only needs to determine a reference point, which is usually set as the worst point of each objective.

5.3. *Parameter setting*

IMOEA has four important parameters: population size N, crossover probability Pc, mutation probability Pm, and neighborhood search scale Ns. Design of experiments (DOE) is performed to decide parameter settings. For each combination in DOE, IMOEA runs 10 times independently. The average value of HV is used as a metric to evaluate each combination and the reference point is set as the worst point of each objective. Finally, the optimal parameter settings of IMOEA are $N = 200$, $Pc = 0.7$, $Pm = 0.1$, and $Ns = 8$. In the same way, DOE is performed on each comparison algorithm to make its parameters set to the best.

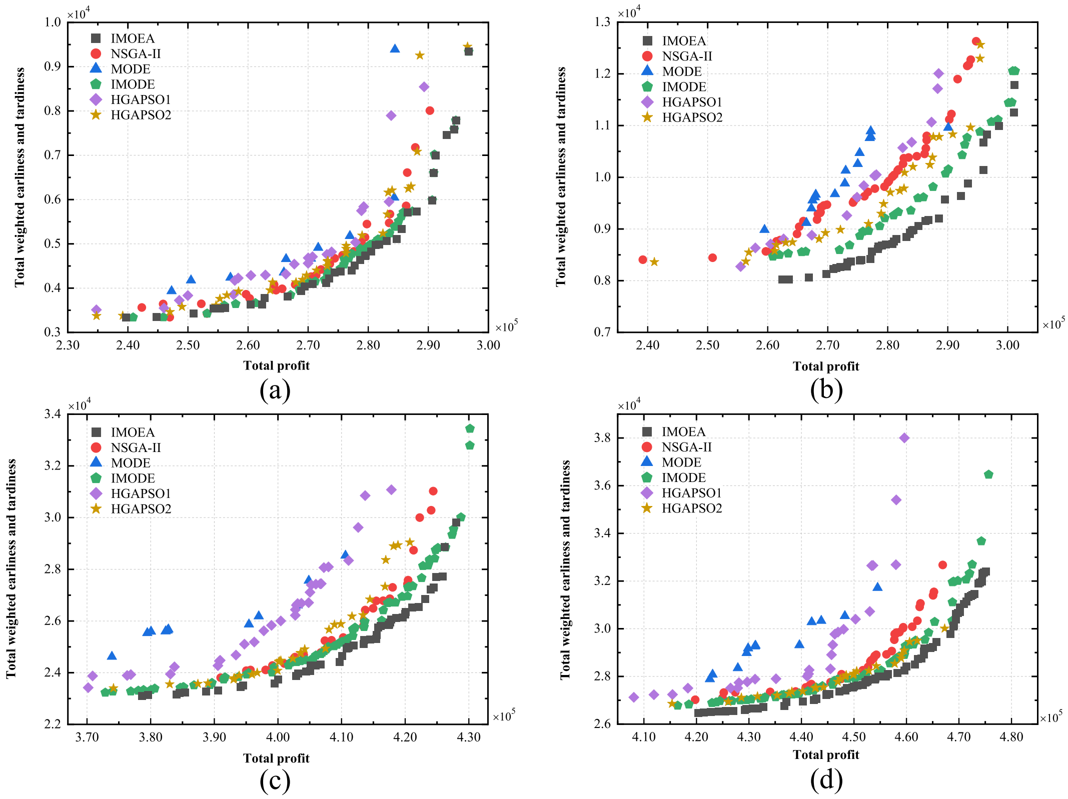
5.4. *Algorithm comparison and result analysis*

To test the performance of IMOEA, we conducted experiments on 30 instances. For avoiding contingency, each algorithm was run 10 times. The experimental results are shown in Table 4, including the best and average values obtained by the algorithm on each objective and the average computation time (time in sec.) for 10 runs. For *f1*, the larger the best value and the average value, the better. And the opposite for *f2*. For the medium- and large-scale instances, IMOEA obtains at least two best performances on each instance, followed by IMODE, and the rest of algorithms obtain very few. With the enlargement of the instance scale, the performance advantage of IMOEA is gradually clear.

For the computation time of the algorithms, NSGA-II and MODE have the shortest computation time, but they also have worse performance. Especially when the instance scale enlarges, they can barely obtain the best performance. Then, for IMODE, HGAPSO1, and HGAPSO2, although their computation time is generally longer due to the addition of some improvement strategies or multi-algorithm hybrid mode in the algorithms, the algorithm performance is improved to a certain extent. Finally, the computation time of IMOEA is similar to IMODE, HGAPSO2, and less than HGAPSO1. However, IMOEA has the best performance and this advantage becomes more obvious as the instance scale enlarges.

Fig. 8 shows the Pareto fronts for all algorithms in solving the four typical instances of different scales. The solutions obtained by IMOEA are closer to the ideal Pareto front. The Pareto front of IMOEA almost completely dominates the other algorithms. Tables 5–6 show the comparison results of six algorithms on the metrics ER, DM, IGD, and HV. The performance of IMOEA is better than the other five algorithms on four metrics.

To further illustrate that the differences between the algorithms are statistically significant. Initially, an Anderson-Darling test was conducted to check the normality of the population formed by each algorithm on 30 instances. Most of the population data had a p-value<0.05. This indicates that the normality assumption is not satisfied, and a non-parametric test is required. Then, the Kruskal-Wallis test of multiple independent populations was performed for the four metrics, the p-value was less than 0.000, and it was concluded that there is sufficient statistical evidence to reject the null hypothesis that all populations are statistically similar. The ranked means are shown in Table 7. IMOEA is significantly better than other algorithms in a rank-based comparison. Finally, in order to explain whether the difference between the two algorithms is significant, the Mann-Whitney test is performed by setting the significance level to 0.05. The corresponding results are listed in Table 8. It is concluded that there are statistically significant differences between IMOEA and all the compared algorithms. Therefore, IMOEA has significant advantages in terms of cardinality, spread, distribution, and convergence.



(a)

(b)

(c)

(d)

Fig. 8. Pareto fronts of six algorithms for (a) 3–4-20–1, (b) 3-5-25–1, (c) 3–5-30–3, and (d) 4–6-30–2.

Table 5
Comparison results of six algorithms on ER and DM.

Scale	Instance	IMOEA		NSGA-II		MODE		IMODE		HGAPSO1		HGAPSO2	
		ER	DM	ER	DM	ER	DM	ER	DM	ER	DM	ER	DM
Small	3–4-20–1	**0.813**	**176.953**	1.000	408.925	1.000	1012.409	0.890	235.798	1.000	478.656	1.000	277.259
	3–4-20–2	**0.753**	**155.178**	0.961	481.209	1.000	1174.424	0.904	212.584	1.000	427.971	0.930	215.868
	3–4-20–3	**0.697**	95.565	1.000	211.224	1.000	727.748	0.894	140.194	1.000	202.033	1.000	**83.425**
	4–4-20–1	**0.758**	**147.508**	0.963	287.447	1.000	1625.489	0.872	240.508	1.000	289.350	0.976	234.979
	4–4-20–2	**0.796**	304.594	1.000	502.316	1.000	1369.500	0.853	307.313	1.000	**261.630**	0.961	441.164
	4–4-20–3	**0.761**	**224.880**	0.978	648.066	1.000	754.734	0.891	281.417	1.000	722.622	0.965	379.974
	5–5-20–1	**0.790**	**124.744**	0.992	198.338	1.000	1138.073	0.972	130.921	1.000	330.818	0.993	237.228
	5–5-20–2	**0.704**	**193.492**	1.000	322.972	1.000	1256.539	0.938	256.788	1.000	249.178	0.994	320.137
	5–5-20–3	**0.888**	**146.166**	1.000	419.891	1.000	1545.250	0.903	185.117	1.000	548.521	1.000	573.111
Medium	3–5-25–1	**0.879**	**111.588**	1.000	314.359	1.000	1167.475	0.984	186.765	1.000	284.015	1.000	492.729
	3–5-25–2	**0.904**	**72.663**	1.000	302.319	1.000	2380.844	0.925	98.211	1.000	686.965	1.000	254.911
	3–5-25–3	**0.885**	**88.100**	1.000	434.573	1.000	1816.810	0.915	96.308	1.000	525.348	1.000	223.109
	4–5-25–1	**0.882**	130.829	1.000	170.641	1.000	2371.371	0.927	**58.451**	1.000	587.368	1.000	264.839
	4–5-25–2	**0.907**	**104.273**	1.000	260.247	1.000	2656.531	0.993	128.680	1.000	354.563	1.000	344.065
	4–5-25–3	0.931	**88.040**	1.000	212.853	1.000	2226.408	**0.922**	104.706	1.000	461.341	1.000	349.414
	5–5-25–1	**0.811**	**106.613**	1.000	297.388	1.000	2036.566	0.935	132.558	1.000	349.160	1.000	247.862
	5–5-25–2	**0.877**	119.144	1.000	419.506	1.000	65535.000	0.940	**100.982**	1.000	340.005	1.000	325.167
	5–5-25–3	0.925	**130.610**	1.000	220.867	1.000	2559.073	**0.909**	136.095	1.000	493.731	1.000	281.568
Large	3–5-30–1	**0.897**	**79.876**	1.000	159.386	1.000	1237.406	0.948	93.312	1.000	167.392	0.998	303.170
	3–5-30–2	**0.906**	**48.233**	1.000	263.667	1.000	2513.031	0.952	98.749	1.000	327.240	1.000	234.979
	3–5-30–3	**0.889**	**67.976**	1.000	185.819	1.000	1510.680	0.955	103.358	1.000	393.756	1.000	221.944
	4–6-30–1	**0.908**	**57.929**	1.000	232.422	1.000	2667.464	0.963	71.596	1.000	256.856	1.000	458.881
	4–6-30–2	**0.860**	**44.443**	1.000	183.751	1.000	2361.161	0.996	294.517	1.000	283.086	1.000	355.926
	4–6-30–3	**0.887**	**55.202**	1.000	139.925	1.000	1186.748	0.983	86.569	1.000	277.500	1.000	292.920
	5–5-30–1	**0.853**	**65.689**	1.000	193.731	1.000	1863.349	0.977	111.173	1.000	313.825	1.000	320.307
	5–5-30–2	**0.891**	**49.909**	1.000	195.271	1.000	2546.834	0.961	102.293	1.000	245.105	1.000	289.495
	5–5-30–3	**0.874**	**46.170**	1.000	159.133	1.000	2680.882	0.968	87.768	1.000	258.836	1.000	274.963
	5–6-30–1	**0.876**	**47.744**	1.000	160.720	1.000	2329.350	0.984	81.334	1.000	258.443	1.000	341.762
	5–6-30–2	**0.850**	**43.333**	1.000	158.242	1.000	1508.620	0.996	98.344	1.000	339.296	1.000	339.869
	5–6-30–3	**0.888**	**44.628**	1.000	166.338	1.000	1631.130	0.994	97.923	1.000	290.847	1.000	271.301

The bold value means the best performance.

Table 6
Comparison results of six algorithms on IGD and HV.

Scale	Instance	IMOEA		NSGA-II		MODE		IMODE		HGAPSO1		HGAPSO2	
		IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
Small	3–4-20–1	**532.072**	**0.834**	2204.650	0.743	4709.996	0.655	675.967	0.827	2659.055	0.721	1529.340	0.780
	3–4-20–2	**570.443**	**0.772**	1733.405	0.711	5120.954	0.552	929.603	0.758	2737.517	0.646	987.322	0.748
	3–4-20–3	**418.533**	**0.817**	1604.668	0.754	5068.178	0.618	740.428	0.803	2844.540	0.695	1169.140	0.778
	4–4-20–1	**555.542**	**0.783**	1554.735	0.715	4905.358	0.565	730.764	0.772	2270.579	0.666	1213.191	0.738
	4–4-20–2	**815.066**	**0.788**	2367.274	0.703	5582.110	0.565	1007.180	0.770	2757.711	0.673	1669.226	0.742
	4–4-20–3	**706.247**	**0.797**	2046.905	0.738	5650.289	0.580	910.848	0.788	2609.101	0.687	1511.968	0.762
	5–5-20–1	**722.923**	**0.781**	2019.750	0.688	5069.829	0.557	939.417	0.757	3171.879	0.622	1434.771	0.728
	5–5-20–2	**829.892**	**0.812**	2373.508	0.723	6537.529	0.573	1187.822	0.788	4019.889	0.649	1693.064	0.759
	5–5-20–3	**763.913**	0.827	1679.411	0.766	5209.076	0.666	775.504	**0.828**	2649.742	0.722	1770.440	0.772
Medium	3–5-25–1	**1007.801**	**0.808**	2765.153	0.674	6332.107	0.561	1700.845	0.759	3772.852	0.629	2342.822	0.707
	3–5-25–2	**841.104**	**0.782**	3132.511	0.653	6829.452	0.511	1121.620	0.763	4551.726	0.587	1856.008	0.721
	3–5-25–3	**908.005**	**0.774**	3187.893	0.644	6745.757	0.501	973.519	0.767	4767.538	0.568	2027.207	0.701
	4–5-25–1	**984.073**	0.749	3138.001	0.646	8019.154	0.464	1055.831	**0.754**	5693.210	0.546	2474.548	0.664
	4–5-25–2	**1026.912**	**0.777**	2849.824	0.676	7495.419	0.491	1312.834	0.760	4091.688	0.608	2526.806	0.689
	4–5-25–3	**952.843**	**0.768**	2665.808	0.684	8236.468	0.511	1164.707	0.766	4186.207	0.619	2383.204	0.691
	5–5-25–1	**1080.870**	**0.755**	3149.937	0.636	8128.502	0.461	1274.571	0.733	4307.272	0.584	2950.091	0.650
	5–5-25–2	**1043.166**	**0.780**	3250.632	0.676	7875.435	0.492	1168.243	0.768	4494.356	0.608	2443.697	0.702
	5–5-25–3	**1116.070**	**0.769**	3503.074	0.653	7184.169	0.478	1460.919	0.763	4042.472	0.595	2376.074	0.684
Large	3–5-30–1	**738.274**	**0.807**	3023.203	0.715	10183.844	0.529	1047.356	0.787	5011.948	0.641	3317.072	0.695
	3–5-30–2	**660.972**	**0.808**	2904.652	0.713	7705.875	0.559	1117.831	0.787	4507.999	0.650	2867.596	0.710
	3–5-30–3	**714.107**	**0.807**	3173.211	0.707	8836.837	0.539	1190.720	0.796	5003.248	0.645	3007.602	0.705
	4–6-30–1	**713.236**	**0.827**	3803.457	0.705	9646.208	0.556	1097.251	0.805	6049.145	0.645	3124.924	0.727
	4–6-30–2	**980.297**	**0.831**	3533.873	0.716	10567.616	0.542	1857.136	0.786	6496.831	0.631	3359.326	0.720
	4–6-30–3	**689.569**	**0.829**	2964.170	0.712	8890.875	0.548	1153.726	0.798	5414.265	0.634	2934.974	0.714
	5–5-30–1	**702.000**	**0.822**	2942.195	0.727	7944.540	0.583	1382.595	0.796	4093.221	0.678	3235.909	0.708
	5–5-30–2	**590.214**	**0.827**	2959.399	0.729	8529.088	0.567	1066.465	0.808	4941.014	0.663	3535.764	0.704
	5–5-30–3	**960.989**	**0.796**	3012.551	0.703	8021.048	0.551	1469.603	0.764	4386.007	0.646	3435.849	0.682
	5–6-30–1	**696.718**	**0.812**	3120.787	0.697	9212.530	0.521	1391.772	0.790	4966.893	0.618	3735.387	0.670
	5–6-30–2	**775.771**	**0.821**	2925.011	0.725	9042.253	0.546	1362.648	0.781	5500.381	0.639	3631.297	0.691
	5–6-30–3	**966.183**	**0.821**	3384.593	0.706	8405.218	0.549	1649.416	0.782	5194.568	0.639	3511.041	0.700

The bold value means the best performance.

Table 7
Ranked means of the Kruskal-Wallis test.

Algorithms	ER	DM	IGD	HV
IMOEA	**18.167**	**28.867**	**19.700**	**156.233**
NSGA-II	116.833	88.000	98.067	84.900
MODE	126.000	165.500	163.800	18.233
IMODE	45.733	44.833	43.800	141.700
HGAPSO1	126.000	113.267	128.600	48.767
HGAPSO2	110.267	102.533	89.067	93.167

Table 8
Computational results of the Mann-Whitney test.

P-value (\: means unable to test)	*P-value* (ER)	*P-value* (DM)	*P-value* (IGD)	*P-value* (HV)
IMOEA- NSGA-II	2.056E-12	2.731E-09	1.510E-11	2.039E-11
IMOEA-MODE	\	1.510E-11	1.510E-11	1.510E-11
IMOEA-IMODE	2.155E-08	1.034E-02	5.968E-07	1.751E-03
IMOEA-HGAPSO1	\	8.066E-11	1.510E-11	1.510E-11
IMOEA-HGAPSO2	4.380E-12	4.878E-10	2.488E-11	1.436E-10
IMODE-MODE	\	1.510E-11	1.510E-11	1.510E-11

As shown in Table 8, IMOEA and IMODE are significantly better than NSGA-II and MODE respectively. In IMOEA, its MOEA is designed based on NSGA-II. Therefore, the proposed ALNS can effectively improve the performance of NSGA-II and MODE. Moreover, by comparing the performance of IMOEA and IMODE, the hybrid of MOEA and ALNS gives a better balance between exploration and exploitation than the hybrid of MODE and ALNS. Fig. 9 shows the box plots of the four metrics. The median of the IMOEA is at the best of the four metrics. Noticeably, for metrics DM, IGD, and HV, the width of the IMOEA box (interquartile range IQR) is small, indicating that the data has low volatility and high stability. Therefore, IMOEA can well address the investigated IPDS problem.

6. Conclusions and future research

In this paper, we present an integrated production scheduling and vehicle routing problem with batch manufacturing and soft time windows. A bi-objective mixed-integer programming model with total profits and total weighted earliness and tardiness is proposed. A hybrid collaborative framework is designed to solve this model, which involves a collaborative mechanism that simultaneously coordinates batch manufacturing and vehicle tour departure schedules. IMOEA is developed according to the proposed hybrid collaborative framework. In IMOEA, the collaborative mechanism is realized by the DPT function and BSS-NWS, which is used as the decoding part. We perform a comparison of the algorithms on instances of different scales. The comparative results of computational experiments show that the efficient cooperation between ALNS and MOEA has made IMOEA's exploration and exploitation capabilities well-balanced. IMOEA performs well in terms of cardinality, convergence, distribution and spread. Therefore, IMOEA is a very competitive method to solve this problem.

For future research, raw material transportation and product delivery vehicles can be used together when considering the supplier stage (Schenekemberg et al., 2021). So how to make a four-stage decision plan that integrates supply, production, inventory, and distribution is an interesting problem. In addition, the selective acceptance and real-time arrival of orders will also bring great challenges to integrated decision-making (An et al., 2023a; An et al., 2023b). Finally, this introduces new complexity to the problem if the travel time for the delivery is uncertain.

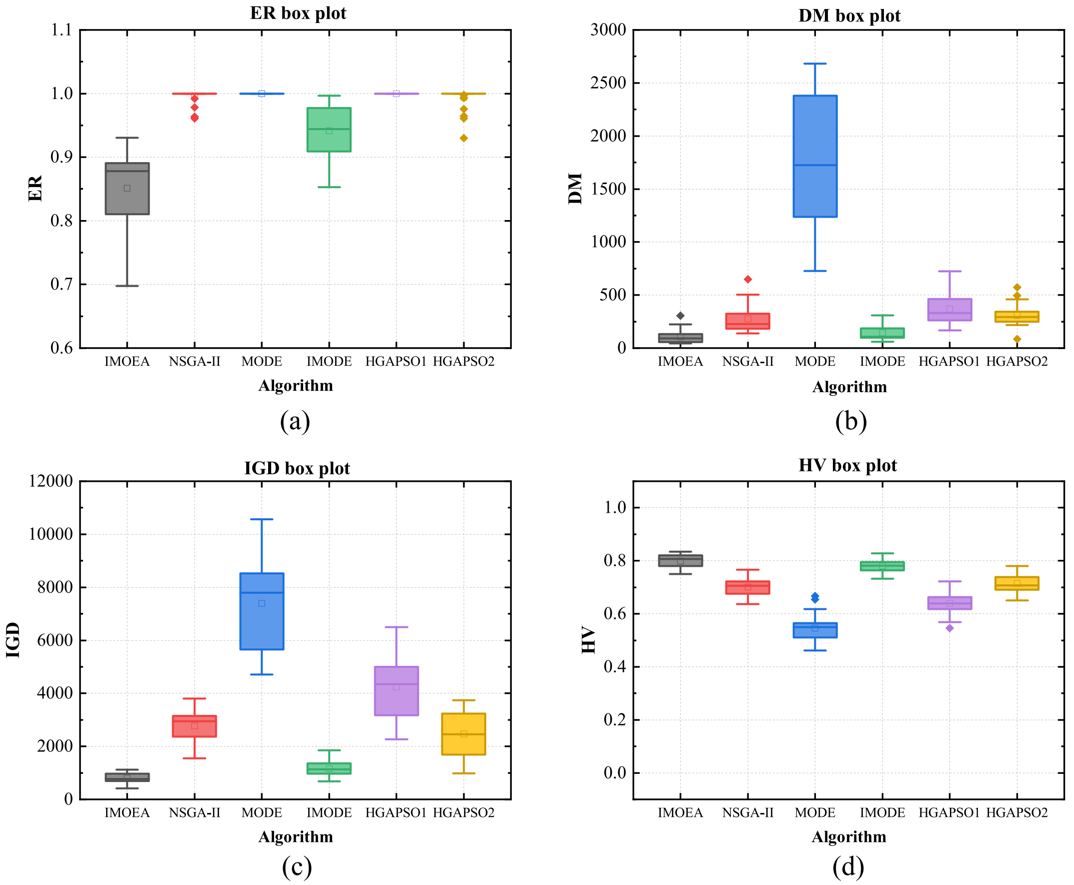


Fig. 9. Box plot of four metrics.

CRediT authorship contribution statement

Ming Huang: Methodology, Writing – original draft. **Baigang Du:** Conceptualization, Investigation, Funding acquisition. **Jun Guo:** Supervision, Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 51705386); China Scholarship Council (No.201606955091).

References

Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., Speranza, M.G., 2018. Comparing sequential and integrated approaches for the production routing problem. European Journal of Operational Research 269 (2), 633–646. https://doi.org/10.1016/j.ejor.2018.01.052.

Aksen, D., Kaya, O., Sibel Salman, F., Tüncel, Ö., 2014. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. European Journal of Operational Research 239 (2), 413–426. https://doi.org/10.1016/j.ejor.2014.05.043.

An, Y., Chen, X., Gao, K., Zhang, L., Li, Y., Zhao, Z., 2023a. A hybrid multi-objective evolutionary algorithm for solving an adaptive flexible job-shop rescheduling problem with real-time order acceptance and condition-based preventive maintenance. Expert Systems with Applications 212, 118711. https://doi.org/10.1016/j.eswa.2022.118711.

An, Y., Chen, X., Gao, K., Zhang, L., Li, Y., Zhao, Z., 2023b. Integrated optimization of real-time order acceptance and flexible job-shop rescheduling with multi-level imperfect maintenance constraints. Swarm and Evolutionary Computation 77, 101243. https://doi.org/10.1016/j.swevo.2023.101243.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., Salomon, L., 2021. Performance indicators in multiobjective optimization. European Journal of Operational Research 292 (2), 397–422. https://doi.org/10.1016/j.ejor.2020.11.016.

Biuki, M., Kazemi, A., Alinezhad, A., 2020. An integrated location-routing-inventory model for sustainable design of a perishable products supply chain network. Journal of Cleaner Production 260, 120842. https://doi.org/10.1016/j.jclepro.2020.120842.

Chevroton, H., Kergosien, Y., Berghman, L., Billaut, J.-C., 2021. Solving an integrated scheduling and routing problem with inventory, routing and penalty costs. European Journal of Operational Research 294 (2), 571–589. https://doi.org/10.1016/j.ejor.2021.02.012.

Coello Coello, C.A., Cruz Cortés, N., 2005. Solving multiobjective optimization problems using an artificial immune system. Genetic Programming and Evolvable Machines 6 (2), 163–190. https://doi.org/10.1007/s10710-005-6164-x.

Darvish, M., Coelho, L.C., 2018. Sequential versus integrated optimization: Production, location, inventory control, and distribution. European Journal of Operational Research 268 (1), 203–214. https://doi.org/10.1016/j.ejor.2018.01.028.

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.

Devapriya, P., Ferrell, W., Geismar, N., 2017. Integrated production and distribution scheduling with a perishable product. European Journal of Operational Research 259 (3), 906–916. https://doi.org/10.1016/j.ejor.2016.09.019.

Figliozzi, M.A., 2010. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transportation Research Part C: Emerging Technologies 18 (5), 668–679. https://doi.org/10.1016/j.trc.2009.08.005.

Fu, L.-L., Aloulou, M.A., Triki, C., 2017. Integrated production scheduling and vehicle routing problem with job splitting and delivery time windows. International Journal of Production Research 55 (20), 5942–5957. https://doi.org/10.1080/00207543.2017.1308572.

Ganji, M., Kazemipoor, H., Hadji Molana, S.M., Sajadi, S.M., 2020. A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. Journal of Cleaner Production 259, 120824. https://doi.org/10.1016/j.jclepro.2020.120824.

Guo, Z., Shi, L., Chen, L., Liang, Y., 2017. A harmony search-based memetic optimization model for integrated production and transportation scheduling in MTO manufacturing. Omega 66, 327–343. https://doi.org/10.1016/j.omega.2015.10.012.

Hou, Y., Fu, Y., Gao, K., Zhang, H., Sadollah, A., 2022. Modelling and optimization of integrated distributed flow shop scheduling and distribution problems with time windows. Expert Systems with Applications 187, 115827. https://doi.org/10.1016/j.eswa.2021.115827.

Kergosien, Y., Gendreau, M., Billaut, J.C., 2017. A Benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. European Journal of Operational Research 262 (1), 287–298. https://doi.org/10.1016/j.ejor.2017.03.028.

Liu, H.T., Guo, Z.X., Zhang, Z.Z., 2021. A hybrid multi-level optimisation framework for integrated production scheduling and vehicle routing with flexible departure time. International Journal of Production Research 59 (21), 6615–6632. https://doi.org/10.1080/00207543.2020.1821927.

Mohammadi, S., Al-e-Hashem, S.M.J.M., Rekik, Y., 2020. An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company. International Journal of Production Economics 219, 347–359. https://doi.org/10.1016/j.ijpe.2019.05.017.

Moons, S., Ramaekers, K., Caris, A., Arda, Y., 2017. Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion. Computers & Industrial Engineering 104, 224–245. https://doi.org/10.1016/j.cie.2016.12.010.

Oladzad-Abbasabady, N., Tavakkoli-Moghaddam, R., Mohammadi, M., Vahedi-Nouri, B., 2023. A bi-objective home care routing and scheduling problem considering patient preference and soft temporal dependency constraints. Engineering Applications of Artificial Intelligence 119, 105829. https://doi.org/10.1016/j.engappai.2023.105829.

Park, Y.-B., Hong, S.-C., 2009. Integrated production and distribution planning for single-period inventory products. International Journal of Computer Integrated Manufacturing 22 (5), 443–457. https://doi.org/10.1080/09511920802527590.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Computers & Operations Research 34 (8), 2403–2435. https://doi.org/10.1016/j.cor.2005.09.012.

Qu, H., Ai, X.-Y., Wang, L., 2020. Optimizing an integrated inventory-routing system for multi-item joint replenishment and coordinated outbound delivery using differential evolution algorithm. Applied Soft Computing 86, 105863. https://doi.org/10.1016/j.asoc.2019.105863.

Schenekemberg, C.M., Scarpin, C.T., Pécora, J.E., Guimarães, T.A., Coelho, L.C., 2021. The two-echelon production-routing problem. European Journal of Operational Research 288 (2), 436–449. https://doi.org/10.1016/j.ejor.2020.05.054.

Soleimani, H., Kannan, G., 2015. A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks. Applied Mathematical Modelling 39 (14), 3990–4012. https://doi.org/10.1016/j.apm.2014.12.016.

Vahdani, B., Niaki, S.T.A., Aslanzade, S., 2017. Production-inventory-routing coordination with capacity and time window constraints for perishable products: Heuristic and meta-heuristic algorithms. Journal of Cleaner Production 161, 598–618. https://doi.org/10.1016/j.jclepro.2017.05.113.

Wang, D., Zhu, J., Wei, X., Cheng, T.C.E., Yin, Y., Wang, Y., 2019. Integrated production and multiple trips vehicle routing with time windows and uncertain travel times. Computers & Operations Research 103, 1–12. https://doi.org/10.1016/j.cor.2018.10.011.

Yagmur, E., Kesen, S.E., 2021. Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: Memetic algorithm and simulated annealing approaches. Computers & Industrial Engineering 161, 107649. https://doi.org/10.1016/j.cie.2021.107649.

Yagmur, E., Kesen, S.E., 2023. Bi-objective coordinated production and transportation scheduling problem with sustainability: formulation and solution approaches. International Journal of Production Research 61 (3), 774–795. https://doi.org/10.1080/00207543.2021.2017054.

Zhao, Z., Chen, X., An, Y., Li, Y., Gao, K., 2023. A property-based hybrid genetic algorithm and tabu search for solving order acceptance and scheduling problem with trapezoidal penalty membership function. Expert Systems with Applications 218, 119598. https://doi.org/10.1016/j.eswa.2023.119598.

Zheng, K., Yang, R.-J., Xu, H., Hu, J., 2016. A new distribution metric for comparing Pareto optimal solutions. Structural and Multidisciplinary Optimization 55 (1), 53–62. https://doi.org/10.1007/s00158-016-1469-3.

Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8 (2), 173–195. https://doi.org/10.1162/106365600568202.