

# Distributed Flexible Assembly Job Shop Scheduling with Collaborative Production and Delivery: Strategy Analysis and a Path-Relinking-Based Solution Method

Jinghe Sun\*, Zhuo Zhang

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China

## Abstract

This paper addresses a collaborative scheduling problem in distributed processing and assembly factories with multiple locations. With the objectives of minimizing the makespan, total tardiness and transportation cost, a distributed flexible assembly job shop scheduling problem with transportation is formulated, and a solution method is developed by adapting the greedy randomized adaptive search procedure, the path relinking mechanism and variable neighborhood descent to a decomposition framework. Its problem-specific elements are a three-segment encoding that represents machine flexibility and operation sequence flexibility, a topological decoding rule that guarantees precedence feasibility without repair, a factory-consistency repair that enforces the transportation constraints inside the decoder, and a whole-job factory-reassignment neighborhood that is the only move able to act directly on transportation cost. Experiments on 42 instances derived from a real case, including a factorial design of 27 instances over order size, factory network and cost structure, show that the proposed method attains a significantly higher hypervolume than MOEA/D and a deep-Q-network-based memetic algorithm. The strategy comparison shows that collaboration mainly acts on logistics: relative to independent production, collaborative production with transportation constraints reduces transportation cost by about 12% while changing the makespan by less than 1%, and the saving doubles when inter-factory transfers are expensive but falls to 2% when delivery costs dominate. Forbidding cross-regional movement of semi-finished jobs never degrades schedule quality relative to unrestricted collaboration and improves it on 36 of the 42 instances, whereas for the largest order books independent production yields better makespan and tardiness within the same computing budget.

**Keywords:** distributed flexible assembly job shop, operation sequence flexibility, path relinking, variable neighborhood descent, multi-objective optimization

Received on 15 September 2026, accepted on 12 October 2026, published on 19 October 2026

Copyright © 2026 Jinghe Sun *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.15132

## 1. Introduction

With the advancement of information platforms and intelligent digital technologies, traditional manufacturing is undergoing significant upgrades. These technologies not only enhance the efficiency of production processes but also pose challenges to production modes and management approaches. Shared manufacturing platforms such as General Electric's

Predix, Siemens' MindSphere and Alibaba's TaoGongChang are implementing distributed production coordination to optimize the global allocation of production resources and supply chain management [1]. Many manufacturers have also adopted multi-factory models to match the distribution of raw materials and customers [2].

\*Corresponding author. Email: [sunjinghe@nuaa.edu.cn](mailto:sunjinghe@nuaa.edu.cn)

Distributed processing and assembly workshops are the production units of such manufacturing systems, and their scheduling methods directly impact production lead times and costs.

In the context of distributed multi-level manufacturing systems with production and sales across multiple locations, the processing and assembly of products and their components are mainly carried out in-house, except for externally purchased or outsourced parts. The operation mode of such systems is illustrated in Figure 1. These factories belong to the same enterprise. Initially, each factory is responsible for receiving customer orders within its region. Subsequently, the production management department formulates material requirement plans and workshop-level production schedules, dispatching orders to component factories. When a factory's orders exceed its short-term production capacity, other factories are permitted to take on part of the production and carry out cross-regional transportation. Consequently, developing a collaborative scheduling strategy that minimizes both production lead times and transportation costs is a critical objective.

The three objectives of this paper correspond to the three managerial concerns on which the collaborative decision acts. The makespan measures the lead time of the whole order book, until the last job reaches its customer, and shows whether collaboration relieves or overloads the assembly stage. Total tardiness measures the fulfillment of customer commitments; it is computed from the delivery-inclusive completion time, because the customer experiences the arrival of the order rather than the end of production, and moving assembly to a farther site trades transportation time against queueing time. Transportation cost measures the logistics consequence of the same decision, comprising both the inter-factory transfer of components and the delivery to the customer. The three objectives conflict: assembling a job at the site nearest to its customer minimizes delivery time and cost but may overload that site and lengthen the makespan, whereas spreading assembly across sites shortens the makespan but adds transfers. They are therefore optimized as a Pareto set rather than as a weighted sum, which would require the planner to price these trade-offs before seeing them.

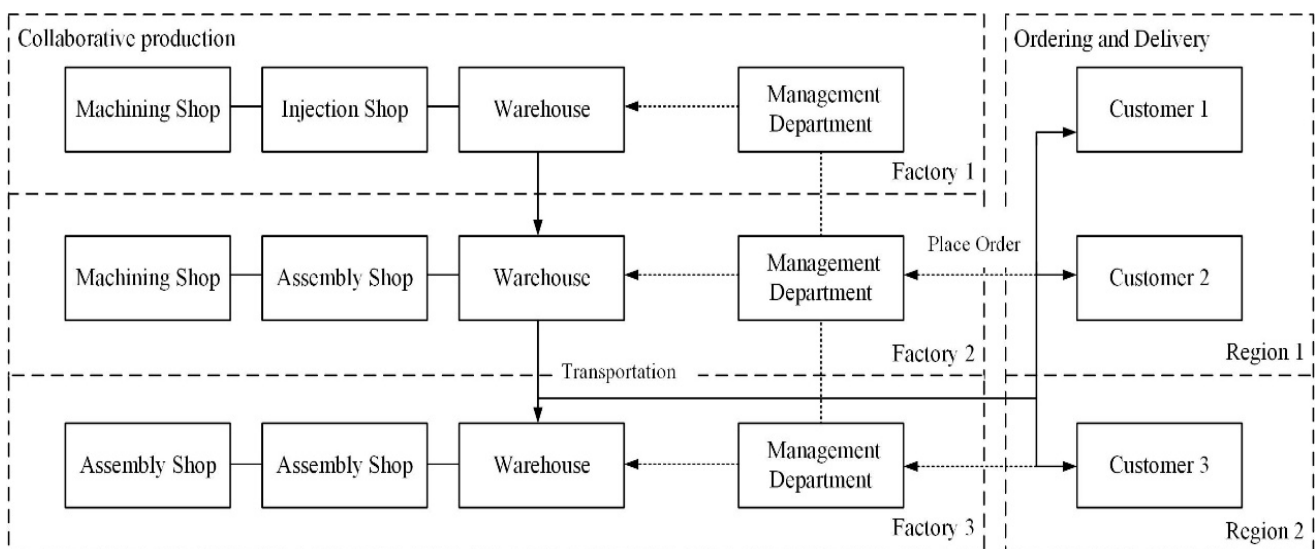


**Figure 1.** Collaborative production and delivery process of a distributed manufacturing system

The flexible job shop scheduling problem (FJSP) has been extensively studied to enhance the production efficiency of flexible manufacturing systems characterized by a variety of products and small batch sizes [3], and mixed-integer formulations with routing and process-plan flexibility are well established [4]. Research relevant to the present study has developed along three strands.

Once assembly is introduced, the operations of a job form a convergent structure rather than a chain, and jobs are no longer mutually independent, which distinguishes the flexible assembly job shop scheduling problem (FAJSP) from the FJSP. Zhang and Wang formalized the FAJSP with sequence-dependent setup times and part sharing, comparing constraint programming, mixed-integer programming and dispatching

rules [5]. Wu et al. addressed the distributed assembly FJSP with an improved differential evolution algorithm [6]. Hu et al. studied the FAJSP in cellular manufacture with a matheuristic-based multi-objective evolutionary algorithm (MMOEA) [7] and later addressed its energy-aware variant [8]. On the distributed axis, the distributed permutation flowshop [9] and its assembly extension [10] made factory assignment part of the scheduling decision; the distributed FJSP has been solved with an improved genetic algorithm [11] and formulated as mixed-integer and constraint programming models [12]; and multi-factory scheduling as a whole has been surveyed in [13, 14]. These works largely develop the assembly axis and the distributed axis separately: the assembly literature typically assumes a single site, while

the distributed literature typically abstracts away the convergent operation structure.

A second strand embeds material handling into the scheduling decision. Transportation constraints between machines were first combined with the FJSP in a single shop [15], and distributed variants have added transportation times between factories [16], transfers of jobs between factories [17] and crane transportation [18]. Homayouni et al. solved the FJSP with transportation by a multistart biased random-key genetic algorithm [19]. Closest to the present problem, Du et al. covered processing, assembly and delivery within a distributed environment and proposed a hybrid estimation of distribution algorithm (HEDA-DEV) minimizing production cost and delivery delay [20]. More broadly, integrated production and outbound distribution scheduling couples production with delivery decisions [21]. What remains open is the selection of the delivery site. When a firm operates several assembly workshops, the choice of which workshop ships to which customer is a transportation-driven decision that feeds back into the production schedule, yet it is normally fixed a priori by customer proximity rather than optimized. Since transportation cost and time are strongly affected by facility selection and routing in multi-origin, multi-destination settings [22], fixing that decision forgoes a substantial share of the achievable savings — and, as Section 4 shows, the direction of the loss is not the one practitioners expect.

Information on related studies is summarized in Table 1, which adds a column classifying the search paradigm of each method. As that column shows, the field is concentrated: almost every method is a population-based evolutionary or swarm algorithm, optionally hybridized with a trajectory-based local search. Within that concentration, the substantive design question is how information is transferred between good solutions. Population-based operators answer it implicitly: crossover recombines parents, and swarm-style movement operators shift an individual towards a better one according to a metaphor-derived rule. The difficulty is that in a constrained encoding space the offspring of two feasible schedules must lie in a restricted region, and an implicit rule gives the designer little control over where in that region the offspring lands or how far it travels. Path relinking provides such control: it has been combined with scatter search for the FJSP [23], and the greedy randomized adaptive search procedure (GRASP) with path relinking has been extended to multi-objective problems [24]; to our knowledge, it has not been adapted to distributed assembly scheduling with transportation. Recent work has responded by adding learning on top of the population, for example the deep-Q-network-enhanced memetic algorithm (DQNMA) for the dual-resource-constrained distributed flexible assembly job shop scheduling problem [25]. Such additions improve adaptivity but leave the underlying information-transfer mechanism implicit.

Table 1. Related works on the flexible assembly job shop scheduling problem

| Reference                     | Production efficiency | Tardiness | Operational cost | Transportation | Method                                                | Search paradigm                            |
|-------------------------------|-----------------------|-----------|------------------|----------------|-------------------------------------------------------|--------------------------------------------|
| Kazemi et al. [2]             | ✓                     | ✓         |                  |                | Two-stage decomposition heuristic                     | Constructive                               |
| Wu et al. [6]                 |                       | ✓         | ✓                | ✓              | Differential evolution + simulated annealing          | Population + trajectory                    |
| Hu et al. [7]                 | ✓                     | ✓         | ✓                |                | Matheuristic-based MOEA (MMOEA)                       | Population + exact                         |
| Hu et al. [8]                 | ✓                     |           | ✓                |                | Artificial bee colony with learning                   | Population + learning                      |
| De Giovanni and Pezzella [11] | ✓                     |           |                  |                | Improved genetic algorithm                            | Population                                 |
| Meng et al. [12]              | ✓                     |           |                  |                | MILP and constraint programming models                | Exact                                      |
| Luo et al. [17]               | ✓                     |           |                  | ✓              | Memetic algorithm                                     | Population + trajectory                    |
| Homayouni et al. [19]         | ✓                     |           |                  | ✓              | Biased random-key genetic algorithm                   | Population                                 |
| Du et al. [20]                | ✓                     | ✓         |                  | ✓              | Estimation of distribution + VNS (HEDA-DEV)           | Population + trajectory                    |
| Zhang et al. [25]             |                       | ✓         | ✓                |                | Deep-Q-network-based memetic algorithm (DQNMA)        | Learning + population                      |
| This paper                    | ✓                     | ✓         | ✓                | ✓              | Decomposition-based GRASP with path relinking and VND | Trajectory + explicit information transfer |

Two gaps follow. First, models covering processing, assembly and delivery still fix the delivery site exogenously, so the differences in production efficiency and logistics cost between near-customer assembly and balanced distributed production have not been quantified. Second, an explicit and

controllable information-transfer mechanism has not been examined for this problem class, even though the encoding of the flexible assembly job shop scheduling problem with transportation (DFAJSP-T) — three segments of heterogeneous type, subject to precedence and factory-

consistency repairs — is precisely the kind of space in which implicit recombination is hard to reason about.

The contributions of this paper are as follows. (1) **Model.** A DFAJSP-T model is formulated in which the assembly and delivery site of each job is a decision rather than a given: processing, assembly and delivery are coupled through a partial precedence set; makespan, total tardiness and transportation cost are optimized simultaneously; and the factory of each operation follows from its machine assignment, so that no explicit factory-allocation variables are needed. (2) **Strategy analysis.** Independent production, unrestricted collaboration and collaboration with transportation constraints are compared on 42 instances derived from a real case, including a factorial design over order size, factory network and cost structure; the comparison quantifies what endogenous site selection buys, when it pays and why, with findings that run against common expectations. (3) **Method.** A solution method is obtained by adapting three established components, the greedy randomized adaptive search procedure (GRASP), path

relinking and variable neighborhood descent (VND), to a decomposition framework. The components are not claimed as contributions; the problem-specific adaptations are, and Section 3 summarizes them in a table. In particular, path relinking between the resident solutions of neighboring weight vectors provides the explicit and controllable information-transfer mechanism whose absence is the second gap.

The remainder of this paper is organized as follows. Section 2 formulates the model, Section 3 presents the solution method, Section 4 reports the experiments and Section 5 concludes.

## 2. Problem formulation

For brevity, the subsequent text collectively refers to the execution of machining and assembly as processing operations. The notation used in the model is explained in Table 2.

Table 2. Description of model-related symbols

| Symbol               | Description                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Indexes              | $i, i'$ Index of jobs, $i=1, 2, \dots, n$                                                                                                                   |
|                      | $j, j'$ Index of operations, $j=1, 2, \dots, q$                                                                                                             |
|                      | $k, k'$ Index of machines, $k=1, 2, \dots, m$                                                                                                               |
|                      | $l$ Index of the customer (delivery destination) of a job                                                                                                   |
|                      | $O_{i,j}$ Operation $j$ of job $i$                                                                                                                          |
| Sets                 | $M_k$ Machine $k$                                                                                                                                           |
|                      | $J, M, O$ Set of all jobs, machines, and operations                                                                                                         |
|                      | $J_i$ Set of operations of job $i$                                                                                                                          |
|                      | $M_{i,j}$ Set of available machines for operation $(i, j)$                                                                                                  |
|                      | $OS$ Set of operation pairs $\{O_{i,j}, O_{i',j'}\}$ in which $O_{i,j}$ must be completed before $O_{i',j'}$ starts                                         |
| Parameters           | $O^A$ Set of assembly operations, whose available machines lie in more than one factory                                                                     |
|                      | $DJ_i$ Delivery deadline of job $i$                                                                                                                         |
|                      | $PT_{i,j,k}$ Processing time of operation $O_{i,j}$ on machine $M_k$                                                                                        |
|                      | $TT_{k,k'}$ Transportation time from the location of $M_k$ to the location of $M_{k'}$ (with $l$ in place of $k'$ : delivery time to the customer)          |
|                      | $UTC_{k,k'}$ Transportation cost from the location of $M_k$ to the location of $M_{k'}$ (with $l$ in place of $k'$ : delivery cost to the customer)         |
| Continuous variables | $A$ A large positive constant                                                                                                                               |
|                      | $C_{max}$ Makespan (maximum completion time of all jobs)                                                                                                    |
|                      | $DT$ Total delivery tardiness                                                                                                                               |
|                      | $Cost$ Total transportation cost                                                                                                                            |
|                      | $S_{i,j}$ Start time of operation $O_{i,j}$                                                                                                                 |
| 0-1 variables        | $CJ_i$ Delivery-inclusive completion time of job $i$ (arrival time at the customer)                                                                         |
|                      | $TJ_i$ Delivery tardiness of job $i$                                                                                                                        |
|                      | $TC_i$ Transportation cost of job $i$                                                                                                                       |
|                      | $RTC_{i,j}$ Transportation cost incurred by operation $O_{i,j}$                                                                                             |
|                      | $x_{i,j,k}$ 1 if operation $O_{i,j}$ is processed on machine $M_k$ , 0 otherwise                                                                            |
| 0-1 variables        | $y_{i,j,i',j'}$ 1 if operation $O_{i,j}$ is processed before $O_{i',j'}$ , 0 otherwise                                                                      |
|                      | $w_{i,j,j',k,k'}$ Auxiliary binary variable for linearization: 1 if $O_{i,j}$ is processed on $M_k$ and its successor $O_{i',j'}$ on $M_{k'}$ , 0 otherwise |

Then, the DFAJSP-T can be described as follows: there are  $n$  jobs  $J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$  to be processed, each

comprising multiple operations  $J_i = \{O_{i,1}, O_{i,2}, \dots, O_{i,j}, \dots, O_{i,q}\}$ , on  $m$  machines  $M =$

$\{M_1, M_2, \dots, M_k, \dots, M_m\}$ . Each machine can handle multiple operations, and operation  $O_{i,j}$  can be executed on any machine within the eligible machine set  $M_{i,j} \subseteq M$ . Some operations are subject to technological precedence constraints, e.g., if  $\langle O_{i,j}, O_{i,j'} \rangle \in OS$ , then  $O_{i,j}$  must be completed before  $O_{i,j'}$ . Beyond the production stage, the DFAJSP-T involves transportation optimization between the factories and customers, which aims to reduce transportation cost and delivery tardiness by optimally allocating customer orders to factories. It is assumed that all jobs to be produced originate from customer orders, so there is a one-to-one correspondence between jobs and delivery locations. Since every machine belongs to a specific factory, the machine assigned to an operation also determines the factory in which it is performed. The scheduling objective is to determine the machine assignment as well as the start and completion times of each operation to minimize the makespan, delivery tardiness and transportation cost. The specific objective functions and constraints are as follows:

$$\min C_{max} \quad (1)$$

$$\min DT = \sum_{i \in J} TJ_i \quad (2)$$

$$\min Cost = \sum_{i \in J} TC_i \quad (3)$$

$$\sum_{M_k \in M_{i,j}} x_{i,j,k} = 1 \quad \forall O_{i,j} \in O \quad (4)$$

$$y_{i,j,i',j'} + y_{i',j',i,j} = 1 \quad \forall O_{i,j}, O_{i',j'} \in O \quad (5)$$

$$y_{i,j,i,j'} = 1 \quad \forall \langle O_{i,j}, O_{i,j'} \rangle \in OS \quad (6)$$

$$S_{i,j} + \sum_{M_k \in M_{i,j}} x_{i,j,k} PT_{i,j,k} + \sum_{M_k \in M_{i,j}} \sum_{M_{k'} \in M_{i,j'}} w_{i,j,j',k,k'} TT_{k,k'} \leq S_{i,j'} \quad \forall \langle O_{i,j}, O_{i,j'} \rangle \in OS \quad (7)$$

$$S_{i,j} + PT_{i,j,k} \leq S_{i,j'} + A(2 - x_{i,j,k} - x_{i,j',k}) + A(1 - y_{i,j,i,j'}) \quad \forall O_{i,j}, O_{i,j'} \in O, \forall M_k \in M_{i,j} \cap M_{i,j'} \quad (8)$$

$$S_{i,q} + \sum_{M_k \in M_{i,q}} x_{i,q,k} PT_{i,q,k} + \sum_{M_k \in M_{i,q}} x_{i,q,k} TT_{k,l} \leq CJ_i \quad \forall i \in J \quad (9)$$

$$Ci_{max} \quad \forall i \quad (10)$$

$$TJ_i \geq CJ_i - DJ_i \quad \forall i \in J \quad (11)$$

$$RTC_{i,j} \geq UTC_{k,k'}(x_{i,j,k} + x_{i,j',k'} - 1) \quad \forall \langle O_{i,j}, O_{i,j'} \rangle \in OS, \forall M_k \in M_{i,j}, M_{k'} \in M_{i,j'} \quad (12)$$

$$TC_i \geq \sum_{O_{i,j} \in J_i} RTC_{i,j} + \sum_{M_k \in M_{i,q}} x_{i,q,k} UTC_{k,l} \quad \forall i \in J \quad (13)$$

$$w_{i,j,j',k,k'} \leq x_{i,j,k} \quad \forall \langle O_{i,j}, O_{i,j'} \rangle \in OS \quad (14)$$

$$w_{i,j,j',k,k'} \leq x_{i,j',k'} \quad \forall \langle O_{i,j}, O_{i,j'} \rangle \in OS \quad (15)$$

$$w_{i,j,j',k,k'} \geq x_{i,j,k} + x_{i,j',k'} - 1 \quad \forall \langle O_{i,j}, O_{i,j'} \rangle \in OS \quad (16)$$

$$S_{i,j}, CJ_i, TJ_i \geq 0 \quad \forall O_{i,j} \in O \quad (17)$$

$$x_{i,j,k}, y_{i,j,i',j'}, w_{i,j,j',k,k'} \in \{0,1\} \quad \forall O_{i,j}, O_{i,j'} \in O, \forall M_k \in M \quad (18)$$

Objective (1) minimizes the makespan; Objective (2) aims to minimize the total delivery tardiness of all jobs; Objective (3) seeks to minimize the sum of all transportation costs from production to delivery for all jobs. Constraints (4)–(18) define the feasible solution space. Constraint (4) describes the decision-making process of the model, which involves assigning a processing machine to each operation while indicating the factory where the job is produced. Constraints (5) and (6) guarantee that there is a unique processing order between any two operations. Constraint (7) enforces the start time constraints between operations of the same job. Constraint (8) enforces the start time constraints between operations on the same machine. Constraint (9) defines the delivery-inclusive completion time of a job, that is, the time at which it reaches its customer. Constraint (10) ensures the makespan is at least as large as the maximum job completion time. Constraint (11) defines the tardiness of job  $i$ . Constraints (12) and (13) represent the transportation cost associated with job  $i$  and its components, including the cost of inter-factory transportation and delivery to customers. Constraints (14)–(16) linearize the product of the machine assignments of consecutive operations through the auxiliary variable  $w_{i,j,j',k,k'}$ . Constraints (17) and (18) specify the ranges of the decision variables.

Note that all  $UTC$ ,  $PT$  and  $TT$  values in the above model are known constants during computation and can be directly retrieved according to their subscripts. The  $UTC_{k,l}$  in Constraint (13) denotes the transportation cost from the location of machine  $k$  to the destination  $l$ , which simply corresponds to the delivery cost of job  $i$ .

In addition, the collaborative production strategy analyzed in Section 4 requires one further constraint, which forbids the cross-regional transportation of semi-finished jobs:

$$x_{i,j,k} + x_{i,j',k'} \leq 1 \quad \forall O_{i,j}, O_{i,j'} \in O^A, \forall M_k \in M_{i,j}, M_{k'} \in M_{i,j'}, TT_{k,k'} > 0 \quad (19)$$

where  $O^A$  denotes the set of assembly operations, that is, those operations whose available machine set spans more than one factory. Since intra-factory transfers incur neither time nor cost,  $TT_{k,k'} > 0$  holds exactly when machines  $M_k$  and  $M_{k'}$  belong to different factories, so Constraint (19) requires all assembly operations of one job to be performed in the same factory. The constraint does not restrict the earlier processing operations, whose available machine sets are confined to a single factory by the equipment layout, so the transfer from the machining workshop to the assembly workshop remains permitted. Note also that both the makespan in Constraint (10) and the delivery tardiness defined by Constraints (9) and (11) are computed from the delivery-inclusive completion time and therefore include the delivery transportation time  $TT_{k,l}$ ; the former measures the lead time of the whole order book and the latter the fulfillment of customer commitments.

### 3. A decomposition-based GRASP with path relinking

The method developed here is a hybrid assembled from established components, and its provenance is stated before the method is described, so that what is adapted and what is new are not confused. GRASP is due to Feo and Resende [26]: each iteration builds a solution by a randomized greedy rule and then improves it by local search, so successive iterations are only weakly correlated. Path relinking is due to Glover [27]: given a starting solution and a guiding solution, it generates a trajectory of intermediate solutions by converting the encoding coordinates in which the two differ, one at a time. Combining the two, so as to reintroduce memory into an otherwise memoryless GRASP, is itself an established design introduced by Laguna and Martí [28]. Variable neighborhood descent is due to Mladenović and Hansen [29]. The decomposition machinery — uniformly distributed weight vectors generated by the method of Das

and Dennis [30], Tchebycheff scalarization, and one resident solution per weight vector updated within a neighborhood of vectors — follows Zhang and Li [31]. Restricting local search to operations on a critical path originates with Nowicki and Smutnicki [32] and was extended to the flexible job shop by Mastrolilli and Gambardella [33].

None of these components is claimed as a contribution. What this section contributes is their adaptation to the DFAJSP-T: the three-segment encoding and its topological decoding rule; the factory-consistency repair that enforces Constraint (19) inside the decoder rather than through a penalty term; the whole-job factory-reassignment neighborhood, which is the only move able to act directly on the transportation cost objective; and the use of path relinking between the resident solutions of neighboring weight vectors as the sole reproduction operator, in place of the crossover and mutation operators normally used in a decomposition framework. Table 3 lists these adaptations, what each does and which difficulty of the DFAJSP-T it addresses.

Table 3. Problem-specific adaptations of the solution method

| Adaptation                                            | What it does                                                                                                                      | Difficulty addressed                                                                                                                                | Section |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Three-segment encoding                                | Separate segments for machine assignment, job sequence and operation keys, each as long as the number of operations               | Routing, sequencing and assembly order are decided jointly, and every individual maps to exactly one schedule                                       | 3.1     |
| Topological decoding                                  | Takes next, among the operations of a job whose predecessors are complete, the one with the smallest key                          | Convergent assembly precedence: any key vector yields a feasible order without post hoc repair                                                      | 3.2     |
| Factory-consistency repair                            | Moves the later assembly operations of a job to the factory of its first one                                                      | Enforces Constraint (19) inside the decoder instead of through a penalty term                                                                       | 3.2     |
| Whole-job factory reassignment ( $N_3$ )              | Moves all assembly operations of a job to another factory at once                                                                 | The only move that changes the delivery factory and thus acts directly on transportation cost; single-operation moves would be undone by the repair | 3.6     |
| Path relinking between neighboring resident solutions | Walks from one resident solution towards that of a neighboring weight vector, committing the best sampled conversion at each step | Explicit and controllable information transfer in place of crossover and mutation; sample size and budget bound its cost                            | 3.5     |

Encoding and decoding rules ensure feasibility even in highly complex solution spaces, but they do not settle how a new schedule should be derived from two that are already good. The useful information carried by such a pair lies in the region of the encoding space between them, and an operator

exploiting it should be able to state how far into that region the new solution is placed and in which direction. An implicit recombination rule does not provide that control, since the position of the offspring is a by-product of the rule rather than a decision. Path relinking supplies exactly that control, as illustrated in Figure 2.

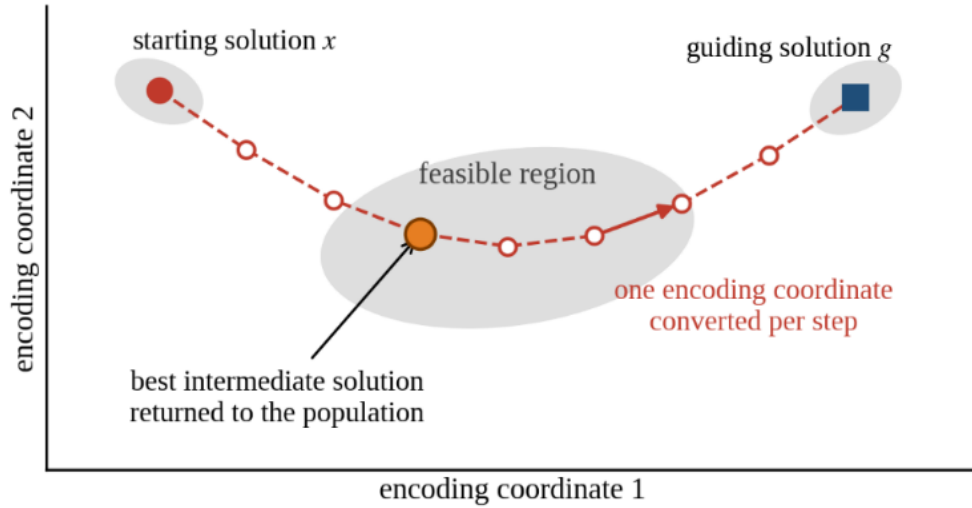


**Figure 2.** Path relinking between a starting solution and a guiding solution in the encoding space

The framework is given in Algorithm 1, with the total number of decoding operations as the termination criterion so that all compared algorithms consume an identical evaluation budget.

**Algorithm 1.** Framework of the decomposition-based GRASP with path relinking

---

# Step

Input: instance data, weight vector divisions  $H$ , neighborhood size  $T$ , decoding budget  $L$

Output: non-dominated solution set  $Nss$

- 1  $Nss \leftarrow \emptyset$ ;  $used \leftarrow 0$
- 2 generate weight vectors  $W$  by the method of Das and Dennis
- 3 build neighborhood index  $B$  by Euclidean distance between weight vectors
- 4 for each  $i \in \{1, \dots, |W|\}$  do
- 5    $pop_i \leftarrow \text{GreedyRandomizedConstruction}()$
- 6    $\text{Decode}(pop_i)$ ;  $used \leftarrow used + 1$
- 7   update ideal point  $z^*$  and nadir point  $z^{nad}$
- 8    $Nss \leftarrow \text{Update}(Nss, pop_i)$
- 9 end for
- 10 while  $used < L$  do
- 11    $i \leftarrow$  next weight vector in cyclic order;  $\lambda \leftarrow W_i$
- 12   if  $\text{rand}() < p_{\text{restart}}$  then
- 13      $x \leftarrow \text{GreedyRandomizedConstruction}()$
- 14   else
- 15      $j \leftarrow \text{RandomSelect}(B_i)$
- 16      $x \leftarrow \text{PathRelinking}(pop_i, pop_j, \lambda)$
- 17   end if

---



---

# Step

- 18  $\text{Decode}(x)$ ;  $used \leftarrow used + \text{number of decodings consumed}$
- 19 update  $z^*$  and  $z^{nad}$
- 20  $x \leftarrow \text{CriticalPathVND}(x, \lambda)$ ;  $used \leftarrow used + \text{number of decodings consumed}$
- 21  $r \leftarrow 0$
- 22 for each  $l \in B_i$  do
- 23   if  $r < 2$  and  $g(x|W_i) < g(pop_l|W_i)$  then
- 24      $pop_l \leftarrow x$ ;  $r \leftarrow r + 1$
- 25   end if
- 26 end for
- 27  $Nss \leftarrow \text{Update}(Nss, \text{all schedules decoded in this iteration})$
- 28 if  $|Nss|$  exceeds its capacity then truncate  $Nss$  by crowding distance
- 29 end while
- 30 return  $Nss$

---

### 3.1 Individual encoding representation

A three-segment encoding is used, each segment having a length equal to the total number of operations, as shown in Figure 3. The machine assignment segment holds, for each operation, the index of the selected machine within its eligible set. The job sequencing segment is a permutation in which job  $i$  appears once per operation. The operation sequencing segment holds a real key in  $(0, 1)$  per operation, whose rank within a job fixes the internal processing order. Each individual thus corresponds uniquely to a schedule.

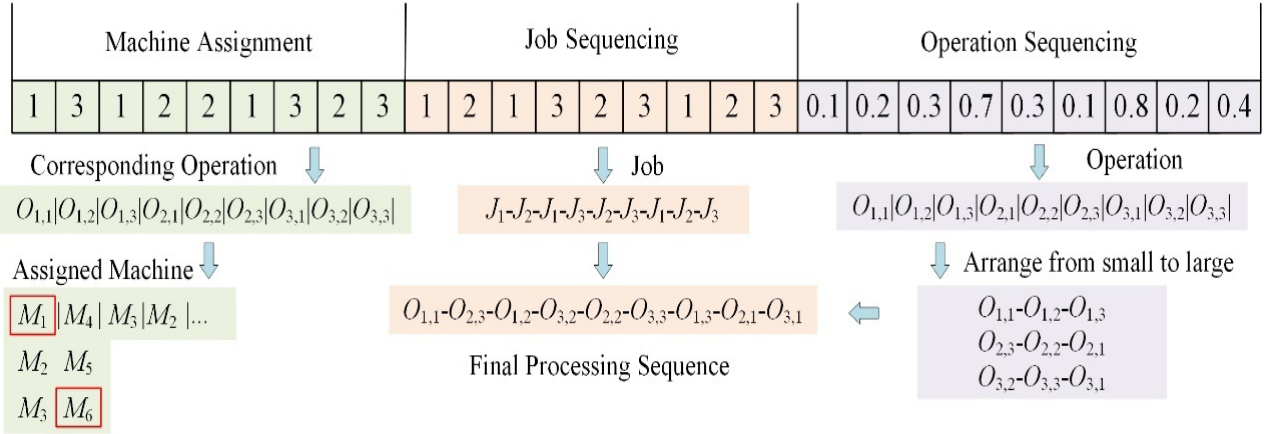


Figure 3. Three-segment encoding and its correspondence to a schedule

### 3.2 Feasible schedule decoding

Decoding proceeds in three passes. First, the internal order of each job is obtained by a topological sort constrained by **OS**: among the operations whose predecessors are all complete, the one with the smallest key is taken next. Any random key vector therefore yields a precedence-feasible order without post hoc repair. Second, the factory-consistency repair of Constraint (19) is applied: for each job, the factory of the first cross-factory-eligible operation is taken as the reference, and any subsequent such operation assigned elsewhere is moved to the eligible machine of the reference factory with the earliest available time; delivery-stage operations are exempt. Removing this pass yields the decoder of the unconstrained collaborative strategy. Third, operations are scheduled in the order given by the job and operation segments using an insertion-based active rule: the release time is the maximum, over all predecessors, of the completion time plus the inter-machine transportation time, and the operation is placed in the earliest idle window of its machine long enough to accommodate it and starting no earlier than the release time; otherwise, it is appended.

**Example.** Table 4 illustrates the decoding of a job  $J_1$  with five operations:  $O_{1,1}$  is machined in factory F1,  $O_{1,2}$  and  $O_{1,3}$  in F2, and the assembly operations  $O_{1,4}$  and  $O_{1,5}$  can be performed in F2 or F3 (machines  $M_1$  and  $M_2$  belong to F1,  $M_3$  to  $M_6$  to F2, and  $M_7$  and  $M_8$  to F3), with **OS** =  $\{\langle O_{1,1}, O_{1,4} \rangle, \langle O_{1,2}, O_{1,3} \rangle, \langle O_{1,3}, O_{1,4} \rangle, \langle O_{1,4}, O_{1,5} \rangle\}$ . Transportation between F1 and F2, F1 and F3, and F2 and F3 takes 2, 5 and 3 time units at costs of 2, 4 and 3, respectively; delivery to the customer takes 2 time units at a cost of 5 from F2 and 3 time units at a cost of 4 from F3; operations of other jobs occupy  $M_1$  during  $[0, 1)$  and  $M_7$  during  $[0, 9)$  and  $[15, 20)$ . In the first pass,  $O_{1,1}$  and  $O_{1,2}$  are available and the smaller key selects  $O_{1,2}$ ;  $O_{1,3}$  then becomes available and precedes  $O_{1,1}$  because of its smaller key, while  $O_{1,4}$  must wait for both of its predecessors although its key is smaller than that of  $O_{1,1}$ . The order is thus  $O_{1,2}, O_{1,3}, O_{1,1}, O_{1,4}, O_{1,5}$ . In the second pass, the first cross-factory-eligible operation,  $O_{1,4}$ , is assigned to  $M_7$  in F3, so F3 becomes the reference factory and  $O_{1,5}$  is moved

from  $M_6$  in F2 to  $M_8$  in F3. In the third pass, the release time of  $O_{1,4}$  is  $\max(6 + 5, 5 + 3) = 11$ , and the operation is inserted into the idle window  $[9, 15)$  of  $M_7$  instead of being appended after 20;  $O_{1,5}$  then runs on  $M_8$  during  $[15, 17)$ . The job reaches its customer at  $CJ_1 = 17 + 3 = 20$  with a transportation cost of  $4 + 3 + 4 = 11$ . Without the repair, as under unrestricted collaboration,  $O_{1,5}$  would stay on  $M_6$ , start after the transfer back to F2 at 18 and reach the customer at 22 with a cost of  $4 + 3 + 3 + 5 = 15$ , which illustrates the kind of transportation-irrational schedule that Constraint (19) removes.

Table 4. Decoding example for one job with five operations

| Op.       | Key  | Eligible   | Encoded | Repaired | PT | Release | Interval   |
|-----------|------|------------|---------|----------|----|---------|------------|
| $O_{1,2}$ | 0.35 | $M_3, M_4$ | $M_4$   | $M_4$    | 3  | 0       | $[0, 3)$   |
| $O_{1,3}$ | 0.18 | $M_3, M_4$ | $M_4$   | $M_4$    | 2  | 3       | $[3, 5)$   |
| $O_{1,1}$ | 0.62 | $M_1, M_2$ | $M_1$   | $M_1$    | 5  | 0       | $[1, 6)$   |
| $O_{1,4}$ | 0.44 | $M_5, M_7$ | $M_7$   | $M_7$    | 4  | 11      | $[11, 15)$ |
| $O_{1,5}$ | 0.90 | $M_6, M_8$ | $M_6$   | $M_8$    | 2  | 15      | $[15, 17)$ |

### 3.3 Randomized greedy construction

Three construction rules are used, one of which is drawn uniformly at random for each construction. Pure randomness fills all three segments at random. Due-date priority ranks jobs by due date minus remaining processing time and draws the next position of the job segment uniformly from a restricted candidate list consisting of the best third of the unfinished jobs, which is what distinguishes a GRASP construction from a deterministic heuristic start. Transportation-cost priority identifies, for each job, the factory with the lowest delivery cost to its customer and draws the machines of its cross-factory-eligible operations from within that factory.

### 3.4 Scalarization and resident solution management

Weight vectors  $\mathbf{W}$  are generated on the standard hyperplane by the method of Das and Dennis [30], and each owns one resident solution. Objectives are normalized by the ideal and nadir points estimated online and aggregated by the Tchebycheff function  $g(\mathbf{z}|\lambda) = \max_k \lambda_k (z_k - z_k^*) / (z_k^{\text{nad}} - z_k^*)$ , which both path relinking and local search use as the acceptance criterion. The use of a scalarizing function rather than a dominance rank is deliberate: in three-objective DFAJSP-T instances a large share of any candidate set is

mutually non-dominated, so a dominance rank supplies little selection pressure. After each iteration the improved schedule may replace the resident solutions of at most two neighboring weight vectors, which propagates good structure locally without letting one schedule take over the whole set, as illustrated in Figure 4. The external archive  $\mathbf{Nss}$  stores the global Pareto set, is updated by every decoded schedule including candidates discarded by local search, and is truncated by crowding distance when it exceeds its cap.

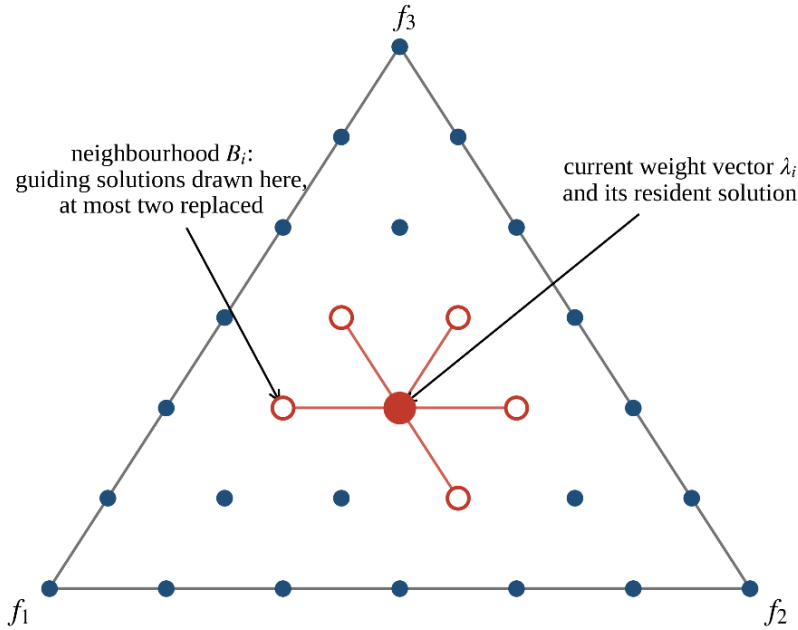


**Figure 4.** Weight vectors, resident solutions and the neighborhood used for replacement

### 3.5 Path relinking

The difference set  $\Delta$  collects every coordinate in which the two encodings disagree: positions of the machine assignment segment with different machine indices, and positions of the operation sequencing segment with different keys. The walk is greedy but sampled. At each step a subset of  $\Delta$  is sampled, the corresponding single-coordinate conversions are applied to the current intermediate schedule and decoded, and the conversion yielding the smallest Tchebycheff value is committed; that coordinate is then removed from  $\Delta$ . Sampling rather than exhaustively evaluating  $\Delta$  bounds the number of decodings a single relinking consumes, which matters because the evaluation budget is the termination criterion. The walk stops when  $\Delta$  is exhausted or the per-relinking budget is spent, and returns the best intermediate schedule rather than either endpoint. Every schedule decoded during the walk is offered to  $\mathbf{Nss}$ , since the corridor between two good schedules frequently contains non-dominated points even when they are poor under the current  $\lambda$ .

### 3.6 Critical-path-guided variable neighborhood descent

Local search alternates among three neighborhood structures, restarting from the first whenever an improvement is found; only improving moves are accepted, and the decodings consumed are charged against the global budget. Which critical chain is traced depends on the objective that the current weight vector emphasizes: for the makespan, the chain is traced backwards from the operation attaining the maximum completion time; for tardiness, from the final operation of the job with the largest delivery-inclusive lateness. Tracing follows at each step either the job predecessor whose completion plus transportation time equals the current start time, or the machine predecessor whose completion equals it.

$N_1$ , critical machine reassignment, changes the machine of an operation on the critical chain to another in its eligible set; restricting the move to the chain is the single most consequential difference from a randomly sampled neighborhood, because reassigning an operation off the chain cannot improve the traced objective.  $N_2$ , critical sequence

shift, moves an occurrence of the critical chain's job earlier in the job segment and perturbs the key of a critical operation.  $N_3$ , whole-job factory move, moves all cross-factory-eligible operations of one job to a single alternative factory at once; it is the only structure that can change the delivery factory, since a single-operation move would be reverted by the factory-consistency repair. Compared with tabu search, this design consumes fewer decodings per improvement, because tabu search must accept non-improving neighbors and maintain a tabu list to escape local optima, spending part of the budget on deliberate degradation.

## 4. Experimental study

### 4.1 Case description

The case originates from the multi-location production system of an optoelectronic equipment manufacturer, which produces radar receivers and laser emitters. As a supplier to several equipment integrators, the firm operates factories in three provinces to secure orders from major clients. The geographical layout and the collaborative relationship among the workshops are shown in Figure 5. The layout comprises an optical workshop in Henan, a central processing workshop and assembly workshop 1 in Hubei, and assembly workshop 2 in Guangdong, with 51 machines in total. Components are processed in the optical and central processing workshops, assembled in either assembly workshop, and then delivered to customers. Both assembly workshops can serve the full range of customers and receive orders directly, with allocation based on proximity to the customer, and the factories operate independently without cross-unit information exchange. To assess the collaborative mode, small, medium and large instances with 5, 10 and 30 jobs, denoted INS-05xx, INS-10xx and INS-30xx, are generated by sampling jobs from the process routes and processing times of the firm's items, with an average of 11 operations per job (upper part of Table 5).

To examine whether the findings extend beyond the case configuration, 27 further instances are generated with the same procedure, which brings the benchmark to 42 instances (Table 5). Jobs are again sampled from the process routes and processing times of the firm's four item types (optical lenses, lens bases, laser emitters and radar receivers), and due dates follow the rule used for the case instances. The additional instances form a full factorial design with three factors at

three levels. The number of jobs is 10, 20 or 40. The factory configuration is the case layout (FC1); an expanded network (FC2), in which a fourth site with eight assembly stations is added, so that every assembly operation can be performed in three factories and every assembled item can be delivered from three sites; or a capacity-reduced layout (FC3), in which each assembly site keeps a single station per assembly stage, so that every assembly operation has two eligible machines instead of four. The transportation cost structure is that of the case (TS1); delivery-dominated (TS2), with all factory-to-customer unit costs tripled; or transfer-dominated (TS3), with all inter-factory unit costs tripled. Transportation times are the same under all three structures, so the cost structure changes the trade-off between delivery and transfer costs without changing which schedules are feasible or how long they take. For each number of jobs the same job set is used in all nine combinations, so that differences between combinations reflect the network rather than the sampled orders. The additional instances are denoted  $En-Fx-Ty$ ; for example, E20-F2-T3 has 20 jobs, the expanded network and the transfer-dominated cost structure. In Table 5, values that differ between the factory configurations are listed in the order FC1 / FC2 / FC3.

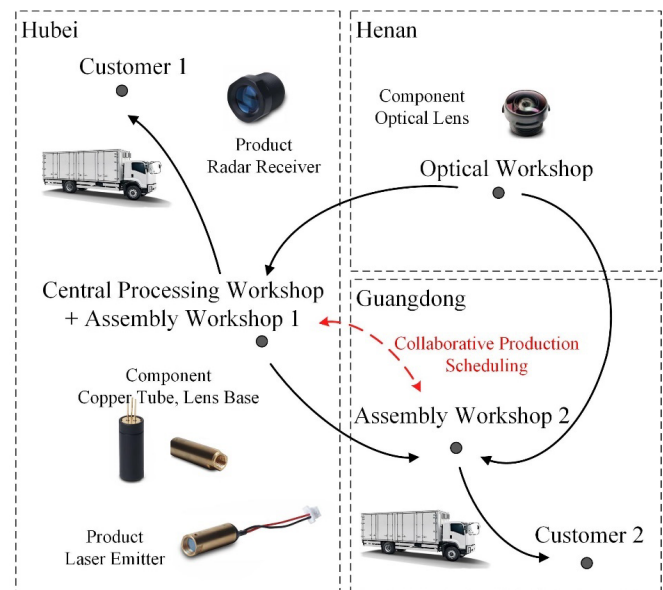


**Figure 5.** Case of collaborative production and delivery across three provinces

**Table 5.** Benchmark instances

| Set      | Instances | Jobs | Factories | Machines | Operations | Cross-factory-eligible operations | Eligible machines per assembly operation | Cost structure |
|----------|-----------|------|-----------|----------|------------|-----------------------------------|------------------------------------------|----------------|
| INS-05xx | 5         | 5    | 3         | 51       | 46–71      | 12–24                             | 4                                        | TS1            |
| INS-10xx | 5         | 10   | 3         | 51       | 107–134    | 31–46                             | 4                                        | TS1            |
| INS-30xx | 5         | 30   | 3         | 51       | 332–361    | 109–123                           | 4                                        | TS1            |

|           |   |    |           |              |     |     |           |               |
|-----------|---|----|-----------|--------------|-----|-----|-----------|---------------|
| E10-Fx-Ty | 9 | 10 | 3 / 4 / 3 | 51 / 59 / 51 | 111 | 37  | 4 / 6 / 2 | TS1, TS2, TS3 |
| E20-Fx-Ty | 9 | 20 | 3 / 4 / 3 | 51 / 59 / 51 | 242 | 79  | 4 / 6 / 2 | TS1, TS2, TS3 |
| E40-Fx-Ty | 9 | 40 | 3 / 4 / 3 | 51 / 59 / 51 | 454 | 153 | 4 / 6 / 2 | TS1, TS2, TS3 |

## 4.2 Experimental settings

The experiments have two parts. The first compares the independent production (IP), collaborative production (CP) and collaborative production with transportation constraints (CPTC) strategies. Under IP the assembly workshop serving a job is fixed to the one with the lowest delivery cost to its customer, so the eligible machines of every cross-factory-eligible operation are confined to a single workshop. CP removes that restriction entirely; CPTC removes it but retains Constraint (19), so that all assembly operations of a job are performed in the same workshop and semi-finished jobs are never moved across regions. The decoder of Section 3.2 implements CPTC directly.

The second part evaluates the proposed method against the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [31] and DQNMA [25]. Convergence is reflected directly by the objective values, and the hypervolume indicator (HV) [34] is used as the comprehensive indicator, a larger value being better. For each instance, objective values are normalized by the ideal and nadir points taken over the union of all solutions produced by all compared methods on that instance, and the reference point is set to (1.1, 1.1, 1.1) in the normalized space. The same normalization is applied within each experiment separately, so that  $HV$  values are comparable across rows of a given table but not across tables, and not with values reported elsewhere under a different reference point. Parameters are listed in Table 6; the maximum number of decodings is the termination criterion and was not tuned. All values in Table 6 were determined empirically in preliminary experiments and fixed before the reported experiments; they are the same for every instance. With  $H = 6$  there are 28 weight vectors; since one iteration consumes at most  $1 + 35 + 90 = 126$  decodings, the budget admits at least 158 iterations, so that every weight vector is revisited at least five times. The remaining parameters decide how this budget is divided between restarts, information transfer by path relinking and intensification by local search. Because the method has many

parameters, no orthogonal tuning design was used; instead, their influence is examined one parameter at a time in Section 4.5, and the 27 additional instances, which played no part in that analysis, serve as an out-of-sample check of the default setting. Each experiment, including those with MOEA/D and DQNMA, was repeated 10 times, and the results are averages; running times are not comparable across implementations and are not reported. All 42 instances, including the 15 case instances, are solved with these settings and the same number of runs. The Wilcoxon rank-sum test detects significant differences in HV at the 0.05 level, with “+”, “−” and “≈” denoting better, worse and indistinguishable results.

Table 6. Algorithm parameter settings

| Main parameter                           | Value          | Local search parameter                                           | Value |
|------------------------------------------|----------------|------------------------------------------------------------------|-------|
| Maximum number of decodings              | 20000          | Neighborhood samples per structure                               | 8     |
| Weight vector divisions $H$              | 6 (28 vectors) | VND decoding budget per call                                     | 90    |
| Neighborhood size $T$                    | 5              | Path relinking budget per call                                   | 35    |
| Restart probability $p_{\text{restart}}$ | 0.10           | Path relinking sample size per step                              | 6     |
| Nss capacity                             | 120            | Maximum replacements per iteration                               | 2     |
|                                          |                | VND neighborhood order ( $N_1 \rightarrow N_2 \rightarrow N_3$ ) | fixed |

## 4.3 Feasibility of collaborative production scheduling

The three strategies are those defined in Section 4.2. The results under the three strategies are shown in Table 7. The models differ only in the set of machines available to each operation. For each instance and strategy,  $HV$  is the mean over 10 runs, and  $C_{\max}$ ,  $DT$  and  $Cost$  are the means over runs of the best value attained in each run’s non-dominated set. Significance marks report the test of each strategy against CPTC on  $HV$ .

Table 7. Comparative scheduling results under the three operating strategies

| Instance | IP     |            |       |        | CP     |            |       |        | CPTC  |            |       |        |
|----------|--------|------------|-------|--------|--------|------------|-------|--------|-------|------------|-------|--------|
|          | $HV$   | $C_{\max}$ | $DT$  | $Cost$ | $HV$   | $C_{\max}$ | $DT$  | $Cost$ | $HV$  | $C_{\max}$ | $DT$  | $Cost$ |
| INS-0501 | 0.049− | 77.0       | 28.1  | 69.0   | 0.768≈ | 77.0       | 27.1  | 67.0   | 0.777 | 77.0       | 27.2  | 67.0   |
| INS-0502 | 0.003− | 79.0       | 94.0  | 86.0   | 0.834≈ | 71.0       | 87.0  | 76.0   | 0.834 | 71.0       | 87.0  | 76.0   |
| INS-0503 | 0.040− | 83.0       | 17.0  | 84.0   | 1.195≈ | 78.4       | 17.0  | 70.0   | 1.181 | 78.8       | 17.0  | 70.0   |
| INS-0504 | 0.011− | 83.0       | 0.0   | 87.0   | 1.195≈ | 75.9       | 0.0   | 63.0   | 1.299 | 75.1       | 0.0   | 63.0   |
| INS-0505 | 0.121− | 76.0       | 99.0  | 70.0   | 1.331≈ | 76.0       | 99.0  | 59.0   | 1.331 | 76.0       | 99.0  | 59.0   |
| INS-1001 | 0.849− | 84.2       | 199.4 | 147.0  | 0.814− | 86.0       | 211.7 | 134.0  | 0.920 | 84.6       | 206.9 | 134.0  |
| INS-1002 | 0.479− | 96.2       | 18.0  | 200.0  | 1.078− | 99.0       | 17.0  | 144.7  | 1.177 | 98.5       | 17.0  | 140.0  |
| INS-1003 | 0.319− | 87.9       | 265.2 | 178.0  | 0.911− | 90.0       | 284.4 | 143.7  | 1.134 | 87.4       | 268.4 | 142.0  |
| INS-1004 | 1.079− | 99.9       | 36.0  | 137.0  | 1.117− | 100.1      | 36.0  | 129.0  | 1.169 | 97.9       | 36.0  | 129.0  |

| Instance          | IP        |                        |           |             | CP        |                        |           |             | CPTC      |                        |           |             |
|-------------------|-----------|------------------------|-----------|-------------|-----------|------------------------|-----------|-------------|-----------|------------------------|-----------|-------------|
|                   | <i>HV</i> | <i>C<sub>max</sub></i> | <i>DT</i> | <i>Cost</i> | <i>HV</i> | <i>C<sub>max</sub></i> | <i>DT</i> | <i>Cost</i> | <i>HV</i> | <i>C<sub>max</sub></i> | <i>DT</i> | <i>Cost</i> |
| INS-1005          | 1.065≈    | 84.0                   | 30.0      | 142.0       | 0.985−    | 86.2                   | 30.0      | 137.0       | 1.081     | 84.5                   | 30.0      | 137.0       |
| INS-3001          | 0.852+    | 159.8                  | 803.0     | 514.0       | 0.623−    | 166.5                  | 908.2     | 501.7       | 0.769     | 162.9                  | 907.1     | 479.5       |
| INS-3002          | 0.935+    | 163.3                  | 574.1     | 481.0       | 0.764−    | 168.6                  | 677.9     | 466.9       | 0.909     | 167.8                  | 626.4     | 454.2       |
| INS-3003          | 0.907−    | 161.2                  | 670.8     | 500.0       | 0.711−    | 167.1                  | 754.1     | 493.9       | 1.010     | 162.9                  | 682.6     | 458.5       |
| INS-3004          | 0.905+    | 159.5                  | 856.4     | 459.0       | 0.670−    | 164.7                  | 964.9     | 448.7       | 0.798     | 164.5                  | 944.0     | 430.3       |
| INS-3005          | 0.963+    | 159.4                  | 820.1     | 477.0       | 0.663−    | 166.3                  | 944.1     | 473.5       | 0.815     | 163.9                  | 904.6     | 454.7       |
| Avg. (case)       | 0.572     | 110.2                  | 300.7     | 242.1       | 0.911     | 111.5                  | 337.2     | 227.2       | 1.014     | 110.2                  | 323.5     | 219.6       |
| E10-F1-T1         | 0.752−    | 97.5                   | 158.0     | 164.0       | 0.986−    | 102.2                  | 165.6     | 138.0       | 1.067     | 100.4                  | 162.7     | 138.0       |
| E10-F1-T2         | 0.984−    | 98.5                   | 157.9     | 298.0       | 1.008≈    | 101.7                  | 164.4     | 281.8       | 1.039     | 101.2                  | 160.9     | 286.0       |
| E10-F1-T3         | 0.515−    | 98.2                   | 156.0     | 358.0       | 0.949−    | 101.9                  | 164.9     | 246.0       | 1.038     | 101.0                  | 160.9     | 246.0       |
| E10-F2-T1         | 0.961−    | 96.8                   | 158.1     | 161.0       | 0.965−    | 99.6                   | 165.8     | 144.3       | 1.091     | 99.7                   | 161.3     | 138.0       |
| E10-F2-T2         | 1.046+    | 96.5                   | 157.5     | 293.0       | 0.945−    | 100.1                  | 166.7     | 283.3       | 0.995     | 99.3                   | 161.0     | 286.2       |
| E10-F2-T3         | 0.694−    | 95.7                   | 155.6     | 351.0       | 0.851−    | 99.5                   | 166.9     | 268.1       | 1.047     | 99.1                   | 157.1     | 246.0       |
| E10-F3-T1         | 0.809−    | 103.0                  | 164.5     | 164.0       | 0.974−    | 107.0                  | 173.0     | 138.9       | 1.047     | 106.9                  | 169.2     | 138.0       |
| E10-F3-T2         | 0.996−    | 101.9                  | 165.1     | 298.0       | 1.000−    | 107.2                  | 169.9     | 282.6       | 1.031     | 106.3                  | 167.8     | 286.0       |
| E10-F3-T3         | 0.593−    | 102.0                  | 164.2     | 358.0       | 0.919−    | 107.2                  | 172.4     | 246.0       | 0.967     | 107.4                  | 170.4     | 246.0       |
| E20-F1-T1         | 0.489−    | 138.2                  | 422.5     | 360.0       | 0.728−    | 137.7                  | 433.6     | 319.1       | 0.935     | 136.0                  | 410.7     | 294.9       |
| E20-F1-T2         | 0.931−    | 137.9                  | 429.9     | 578.0       | 0.884−    | 137.9                  | 452.1     | 562.9       | 1.033     | 135.6                  | 420.5     | 562.3       |
| E20-F1-T3         | 0.340−    | 137.8                  | 410.2     | 862.0       | 0.661−    | 137.4                  | 435.9     | 669.4       | 0.856     | 136.2                  | 427.6     | 559.9       |
| E20-F2-T1         | 0.572−    | 136.2                  | 416.4     | 372.0       | 0.677−    | 132.2                  | 392.5     | 355.9       | 0.956     | 130.5                  | 376.7     | 322.6       |
| E20-F2-T2         | 0.759−    | 137.6                  | 426.8     | 588.0       | 0.836−    | 131.5                  | 407.1     | 583.6       | 1.041     | 130.3                  | 381.5     | 565.4       |
| E20-F2-T3         | 0.370−    | 136.1                  | 424.3     | 900.0       | 0.627−    | 130.8                  | 395.7     | 778.1       | 0.900     | 129.8                  | 384.6     | 660.9       |
| E20-F3-T1         | 0.504−    | 175.8                  | 558.8     | 360.0       | 0.879−    | 157.9                  | 562.9     | 320.7       | 1.017     | 157.4                  | 531.4     | 301.4       |
| E20-F3-T2         | 0.655−    | 175.2                  | 556.8     | 578.0       | 0.890−    | 158.8                  | 568.9     | 559.9       | 1.014     | 156.2                  | 528.1     | 560.0       |
| E20-F3-T3         | 0.353−    | 176.0                  | 556.8     | 862.0       | 0.738−    | 158.0                  | 537.6     | 691.4       | 0.908     | 157.3                  | 535.5     | 567.9       |
| E40-F1-T1         | 1.027+    | 191.2                  | 1742.6    | 668.0       | 0.612−    | 198.0                  | 1911.6    | 695.1       | 0.812     | 195.3                  | 1868.5    | 661.9       |
| E40-F1-T2         | 1.171+    | 192.3                  | 1692.5    | 1220.0      | 0.677−    | 197.8                  | 1893.2    | 1247.0      | 0.891     | 193.7                  | 1841.9    | 1224.4      |
| E40-F1-T3         | 1.027+    | 191.2                  | 1703.6    | 1452.0      | 0.678−    | 198.4                  | 1889.2    | 1501.7      | 0.871     | 196.2                  | 1876.2    | 1379.4      |
| E40-F2-T1         | 1.001+    | 179.3                  | 1584.0    | 735.0       | 0.577−    | 183.8                  | 1719.6    | 764.4       | 0.855     | 181.3                  | 1667.0    | 721.0       |
| E40-F2-T2         | 1.075+    | 178.8                  | 1585.2    | 1257.0      | 0.616−    | 183.0                  | 1697.3    | 1300.5      | 0.828     | 181.4                  | 1660.0    | 1263.7      |
| E40-F2-T3         | 0.839≈    | 178.8                  | 1572.4    | 1683.0      | 0.507−    | 185.6                  | 1732.4    | 1720.6      | 0.789     | 181.9                  | 1666.3    | 1546.6      |
| E40-F3-T1         | 1.094+    | 248.0                  | 2223.7    | 668.0       | 0.757−    | 249.1                  | 2497.3    | 697.7       | 0.930     | 246.3                  | 2435.1    | 671.6       |
| E40-F3-T2         | 1.134+    | 249.1                  | 2261.6    | 1220.0      | 0.777−    | 247.1                  | 2473.3    | 1257.6      | 0.933     | 245.8                  | 2445.7    | 1230.8      |
| E40-F3-T3         | 1.062+    | 248.8                  | 2281.2    | 1452.0      | 0.792−    | 248.3                  | 2497.0    | 1528.2      | 0.999     | 245.0                  | 2406.7    | 1424.5      |
| Avg. (additional) | 0.806     | 151.8                  | 825.4     | 676.3       | 0.797     | 151.8                  | 889.1     | 651.2       | 0.959     | 150.3                  | 864.3     | 612.2       |

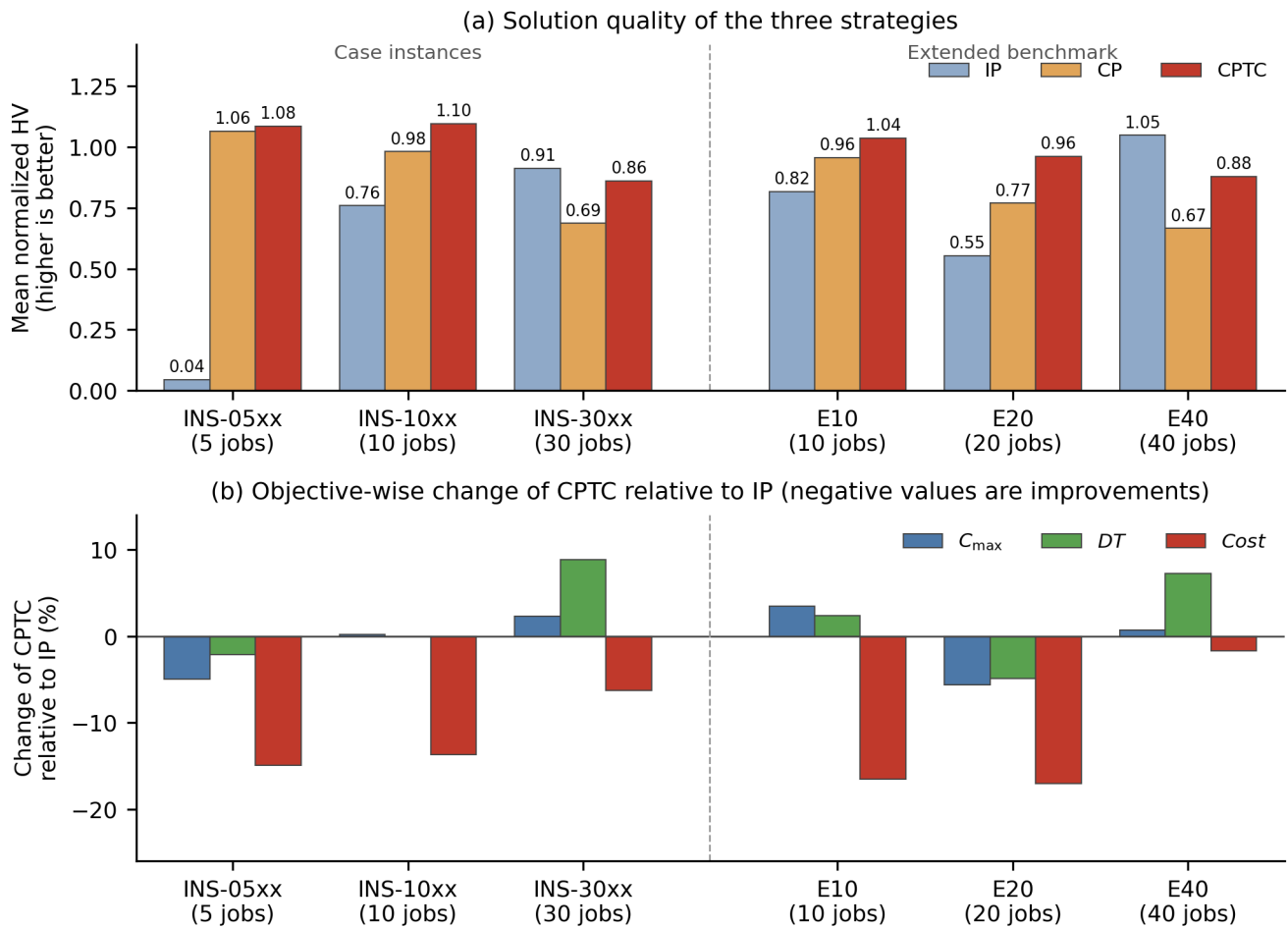


**Figure 6.** Solution quality of the three strategies and the objective-wise change of CPTC relative to IP

Figure 6 summarizes the same results by instance group, together with the three job-count groups of the extended benchmark, contrasting the solution quality of the three strategies with the objective-wise change of CPTC relative to IP. Averaged over the 15 case instances, CPTC attains the highest  $HV$  (1.014), ahead of CP (0.911) and IP (0.572), and also the lowest transportation cost (219.6 against 227.2 and 242.1). The advantage is concentrated in one objective and in the small and medium instances.

**Collaboration is a logistics lever, not a throughput lever.** Relative to IP, CPTC reduces transportation cost on average by 14.9% on the small instances, 13.7% on the medium instances and 6.3% on the large instances (11.6% over all 15 instances), while the makespan changes by only  $-0.8\%$  and the total tardiness by  $+2.2\%$ . The asymmetry follows from the structure of the case: only assembly operations are cross-factory eligible, so collaboration redistributes assembly and delivery work but leaves the processing bottleneck untouched, and it is that bottleneck which determines  $C_{max}$ . Firms evaluating whether to activate cross-factory production should expect the return in the logistics account, not in throughput.

**Collaboration reduces transportation cost, contrary to the usual expectation.** Conventional practice assigns each job to the assembly workshop nearest its customer on the grounds that this minimizes delivery cost. Fixing the workshop by customer proximity optimizes the delivery term of Constraints (12) and (13) while ignoring the inter-factory term, and the arc from the processing workshop to the assembly workshop is often the more expensive of the two. When the saving on that arc exceeds the extra delivery cost, moving the job elsewhere lowers its total cost, and IP forbids exactly that substitution. The extended benchmark confirms this mechanism: tripling the inter-factory unit costs (TS3) raises the saving of CPTC relative to IP from 10.7% to 22.6%, whereas tripling the delivery costs (TS2) reduces it to 2.0% (Table 8).

Beyond the aggregate comparison, it is worth asking where inside the collaborative region the good schedules lie. For every non-dominated solution obtained under CPTC we computed the off-site production ratio ( $OSPR$ ), the proportion of jobs whose assembly operations are performed in a factory other than the one with the lowest delivery cost to their customers; it is read directly from the machine assignment segment and equals zero for every IP solution. Pooled over

10 runs, the mean *OSPR* is 0.347, 0.226 and 0.152 on the small, medium and large case instances and 0.281, 0.345 and 0.151 on the extended instances with 10, 20 and 40 jobs, while on a single instance it ranges from 0 to 0.67. The optimum is therefore interior but not constant: between roughly one and four jobs in ten are better assembled away from their nearest factory, and the share is lowest on the largest instances, where the cost saving of collaboration is also smallest. The off-site share is thus an outcome to be optimized for each order book rather than a fixed planning rule.

**The scale effect runs against the expectation that a larger solution space is better.** On the small and medium instances IP is significantly worse than CPTC on *HV* in 9 of the 10 cases, while CP is level with CPTC on the small instances but significantly worse on all medium ones. On the 30-job instances the ordering changes: IP reaches a mean *HV* of 0.912 against 0.860 for CPTC and is significantly better on 4 of the 5 instances, whereas CP falls to 0.686; the 40-job instances of the extended benchmark repeat this reversal, IP being significantly better on 8 of 9. The case data explain why the reversal appears only on the largest order books. Assembling a job at the Hubei site, which is co-located with the central processing workshop, shortens the inbound transfer by more than it lengthens delivery. For 30–54% of the assembled jobs in each instance group Hubei is therefore cheaper than the site nearest the customer, by 6.6–13.7 cost units per job on average, and with unlimited capacity it would also complete these jobs 2.8–4.7 time units earlier; 98% of the cost-saving relocations point to Hubei. Whether they pay off depends on the assembly load there. On the 30- and 40-job instances IP already gives Hubei more than half of the assembly work (54% and 59%), and moving every cost-saving job there would raise the pure processing load of its busiest assembly machine from 43% to 73% and from 56% to 92% of the IP makespan, and to 118% in the capacity-reduced configuration FC3. The transportation saving is then paid for in queueing at Hubei, that is, in makespan and tardiness: on the 30-job instances CPTC is on average 2.3% and 8.9% worse than IP on these two objectives while its transportation cost is 6.3% lower. On the 20-job instances IP leaves Hubei only 30% of the assembly work, so the same relocations balance the two sites and improve cost and time together, and on smaller order books the assembly lines have ample slack. A search effect compounds this capacity effect. Every IP schedule is also feasible under CPTC, so an exhaustive search could never rank IP higher; under the fixed

budget, however, CPTC does not recover IP-quality schedules, and even the schedules on its fronts that keep every job at the nearest site are worse than those found under IP, presumably because part of the budget is spent on the workshop decision. CP, which prunes nothing, falls behind both.

**The transportation constraints eliminate poor solutions rather than attractive ones.** CPTC is never significantly worse than CP on *HV* and is significantly better on 36 of the 42 instances; on the remaining 6 (the five small case instances and E10-F1-T2) the two strategies are statistically indistinguishable. This supports reading Constraint (19) as a pruning device rather than a restriction on quality. Its benefit grows with the size of the search: the mean *HV* gap between CPTC and CP widens from 0.02 on the small case instances to 0.17 on the large ones, and from 0.08 to 0.21 between the 10-job and 40-job extended instances; on the 30-job instances CPTC also lowers total tardiness by 4.7% and transportation cost by 4.5% relative to CP. The operating rule forbidding cross-regional movement of semi-finished jobs, usually adopted for reasons of handling risk rather than optimality, therefore costs nothing in schedule quality and in fact helps at realistic problem sizes.

**Results on the extended benchmark.** Table 8 confirms the case findings on the additional instances and qualifies them by size; per-instance results are included in Tables 7 and 9. In Table 8, W/T/L counts the instances on which CPTC (the proposed method) is significantly better than, indistinguishable from or worse than IP (the compared algorithm), and  $\Delta C_{\max}$  and  $\Delta Cost$  are the mean changes of the best values under CPTC relative to IP. CPTC attains the highest mean *HV* at every level of the factory-configuration and cost-structure factors and is significantly better than CP on 26 of the 27 instances and never worse. Against IP it is significantly better on 17 of the 18 instances with 10 or 20 jobs, but on the 40-job instances IP is significantly better on 8 of 9, repeating the reversal observed on the 30-job case instances. The transportation-cost saving of CPTC relative to IP averages 11.8% and depends on the cost structure as the mechanism described above predicts: it is 10.7% under the case costs, rises to 22.6% when inter-factory costs are tripled, and shrinks to 2.0% when delivery costs are tripled, in which case assignment to the nearest factory is almost cost-optimal and IP is significantly better than CPTC on 4 of the 9 instances. The saving is similar across the factory configurations (11.1–12.5%) and falls to 1.7% at 40 jobs, where the search of the collaborative space is most stretched.

Table 8. Results on the additional instances by factor level

| Factor | Level | Strategies (proposed method) |       |       |                  |                       | Algorithms under CPTC |        |       |          |              |             |
|--------|-------|------------------------------|-------|-------|------------------|-----------------------|-----------------------|--------|-------|----------|--------------|-------------|
|        |       | IP                           | CP    | CPTC  | CPTC vs IP W/T/L | $\Delta C_{\max}$ (%) | $\Delta Cost$ (%)     | MOEA/D | DQNMA | Proposed | vs           |             |
|        |       |                              |       |       |                  |                       |                       |        |       |          | MOEA/D W/T/L | DQNMA W/T/L |
| Jobs   | 10    | 0.817                        | 0.955 | 1.036 | 8/0/1            | +3.5                  | −16.5                 | 1.082  | 1.071 | 1.088    | 0/9/0        | 2/7/0       |
|        | 20    | 0.553                        | 0.769 | 0.962 | 9/0/0            | −5.6                  | −17.0                 | 0.877  | 0.897 | 0.915    | 3/6/0        | 2/7/0       |

|                       |              |       |       |       |        |      |       |       |       |       |        |        |
|-----------------------|--------------|-------|-------|-------|--------|------|-------|-------|-------|-------|--------|--------|
|                       | 40           | 1.048 | 0.666 | 0.879 | 0/1/8  | +0.7 | -1.7  | 0.836 | 0.877 | 0.916 | 6/3/0  | 3/5/1  |
| Factory configuration | FC1          | 0.804 | 0.798 | 0.949 | 6/0/3  | +1.1 | -12.5 | 0.933 | 0.940 | 0.958 | 2/7/0  | 2/6/1  |
|                       | FC2          | 0.813 | 0.733 | 0.945 | 5/1/3  | 0.0  | -11.1 | 0.916 | 0.931 | 0.965 | 3/6/0  | 3/6/0  |
|                       | FC3          | 0.800 | 0.858 | 0.983 | 6/0/3  | -2.5 | -11.7 | 0.947 | 0.974 | 0.996 | 4/5/0  | 2/7/0  |
| Cost structure        | TS1          | 0.801 | 0.795 | 0.968 | 6/0/3  | -0.4 | -10.7 | 0.896 | 0.922 | 0.945 | 4/5/0  | 1/8/0  |
|                       | TS2          | 0.972 | 0.848 | 0.978 | 5/0/4  | -0.8 | -2.0  | 0.975 | 0.996 | 1.050 | 5/4/0  | 5/4/0  |
|                       | TS3          | 0.644 | 0.747 | 0.931 | 6/1/2  | -0.2 | -22.6 | 0.925 | 0.927 | 0.924 | 0/9/0  | 1/7/1  |
| All                   | 27 instances | 0.806 | 0.797 | 0.959 | 17/1/9 | -0.5 | -11.8 | 0.932 | 0.948 | 0.973 | 9/18/0 | 7/19/1 |

**When CPTC should be preferred in practice.** For a planner, the results translate into four operating guidelines. First, for small and medium order books CPTC should replace nearest-factory assignment wherever assembly capacity exists in more than one region: it is significantly better than IP on *HV* in 26 of the 28 instances with up to 20 jobs and lowers transportation cost on average by 13.7–17.0% in these instance groups, while the makespan changes by between -5.6% and +3.5%. Second, CPTC should always be preferred to unrestricted collaboration, which is never better and is significantly worse on 36 of the 42 instances. Third, the saving is largest where the transfer from the machining site to the assembly site is the expensive leg of the logistics chain: tripling inter-factory unit costs raises it from 10.7% to 22.6%, whereas tripling delivery costs reduces it to 2.0%, in which case nearest-factory assignment is almost cost-optimal; the saving is robust to the factory network (11.1–12.5%). Fourth, for large order books (30 or 40 jobs) under a fixed computing budget, IP yields better makespan and tardiness and a significantly higher *HV* on 12 of 14 instances, because the cost-saving relocations would crowd the assembly lines next to the processing workshop, while CPTC retains a transportation-cost advantage of 1.7–6.3%; collaboration then pays only if transportation cost is the planner's priority or more computing time can be spent on the schedule.

#### 4.4 Algorithm performance evaluation

Table 9 reports the mean *HV* of MOEA/D, DQNMA and the proposed method under the CPTC strategy, with significance marks against the proposed method, and Figure 7 shows the distribution over all runs.

Table 9. Comprehensive performance comparison of the compared algorithm

| Instance | MOEA/D  | DQNMA   | Proposed |
|----------|---------|---------|----------|
| INS-0501 | 0.868 – | 0.938 ≈ | 0.935    |

|                      |         |         |       |
|----------------------|---------|---------|-------|
| INS-0502             | 0.158 ≈ | 0.158 ≈ | 0.158 |
| INS-0503             | 1.140 – | 1.126 – | 1.240 |
| INS-0504             | 0.489 – | 1.080 ≈ | 1.227 |
| INS-0505             | 1.331 ≈ | 1.206 ≈ | 1.331 |
| INS-1001             | 0.968 ≈ | 0.940 ≈ | 0.954 |
| INS-1002             | 1.144 ≈ | 1.173 ≈ | 1.171 |
| INS-1003             | 0.894 – | 0.933 – | 1.052 |
| INS-1004             | 1.107 – | 1.173 ≈ | 1.149 |
| INS-1005             | 1.070 – | 1.108 ≈ | 1.130 |
| INS-3001             | 0.881 – | 0.937 – | 0.988 |
| INS-3002             | 0.915 – | 0.926 – | 0.991 |
| INS-3003             | 0.840 – | 0.921 – | 1.030 |
| INS-3004             | 0.840 ≈ | 0.851 ≈ | 0.890 |
| INS-3005             | 0.845 ≈ | 0.820 – | 0.881 |
| Average (case)       | 0.899   | 0.953   | 1.009 |
| E10-F1-T1            | 1.101 ≈ | 1.043 ≈ | 1.068 |
| E10-F1-T2            | 1.117 ≈ | 1.091 ≈ | 1.089 |
| E10-F1-T3            | 1.028 ≈ | 1.036 ≈ | 1.037 |
| E10-F2-T1            | 1.020 ≈ | 1.034 – | 1.071 |
| E10-F2-T2            | 1.126 ≈ | 1.104 ≈ | 1.108 |
| E10-F2-T3            | 1.052 ≈ | 1.019 – | 1.063 |
| E10-F3-T1            | 1.086 ≈ | 1.081 ≈ | 1.102 |
| E10-F3-T2            | 1.165 ≈ | 1.169 ≈ | 1.185 |
| E10-F3-T3            | 1.047 ≈ | 1.060 ≈ | 1.067 |
| E20-F1-T1            | 0.850 ≈ | 0.875 ≈ | 0.877 |
| E20-F1-T2            | 0.938 ≈ | 0.904 – | 0.970 |
| E20-F1-T3            | 0.868 ≈ | 0.882 ≈ | 0.861 |
| E20-F2-T1            | 0.809 ≈ | 0.844 ≈ | 0.855 |
| E20-F2-T2            | 0.868 – | 0.932 ≈ | 0.975 |
| E20-F2-T3            | 0.910 ≈ | 0.879 ≈ | 0.885 |
| E20-F3-T1            | 0.859 – | 0.895 ≈ | 0.912 |
| E20-F3-T2            | 0.952 – | 0.978 – | 1.034 |
| E20-F3-T3            | 0.844 ≈ | 0.882 ≈ | 0.865 |
| E40-F1-T1            | 0.789 – | 0.838 ≈ | 0.871 |
| E40-F1-T2            | 0.853 – | 0.919 – | 1.026 |
| E40-F1-T3            | 0.854 ≈ | 0.870 + | 0.822 |
| E40-F2-T1            | 0.746 – | 0.791 ≈ | 0.836 |
| E40-F2-T2            | 0.828 – | 0.903 – | 1.025 |
| E40-F2-T3            | 0.885 ≈ | 0.875 ≈ | 0.867 |
| E40-F3-T1            | 0.806 – | 0.894 ≈ | 0.910 |
| E40-F3-T2            | 0.926 – | 0.965 – | 1.041 |
| E40-F3-T3            | 0.835 ≈ | 0.842 ≈ | 0.846 |
| Average (additional) | 0.932   | 0.948   | 0.973 |

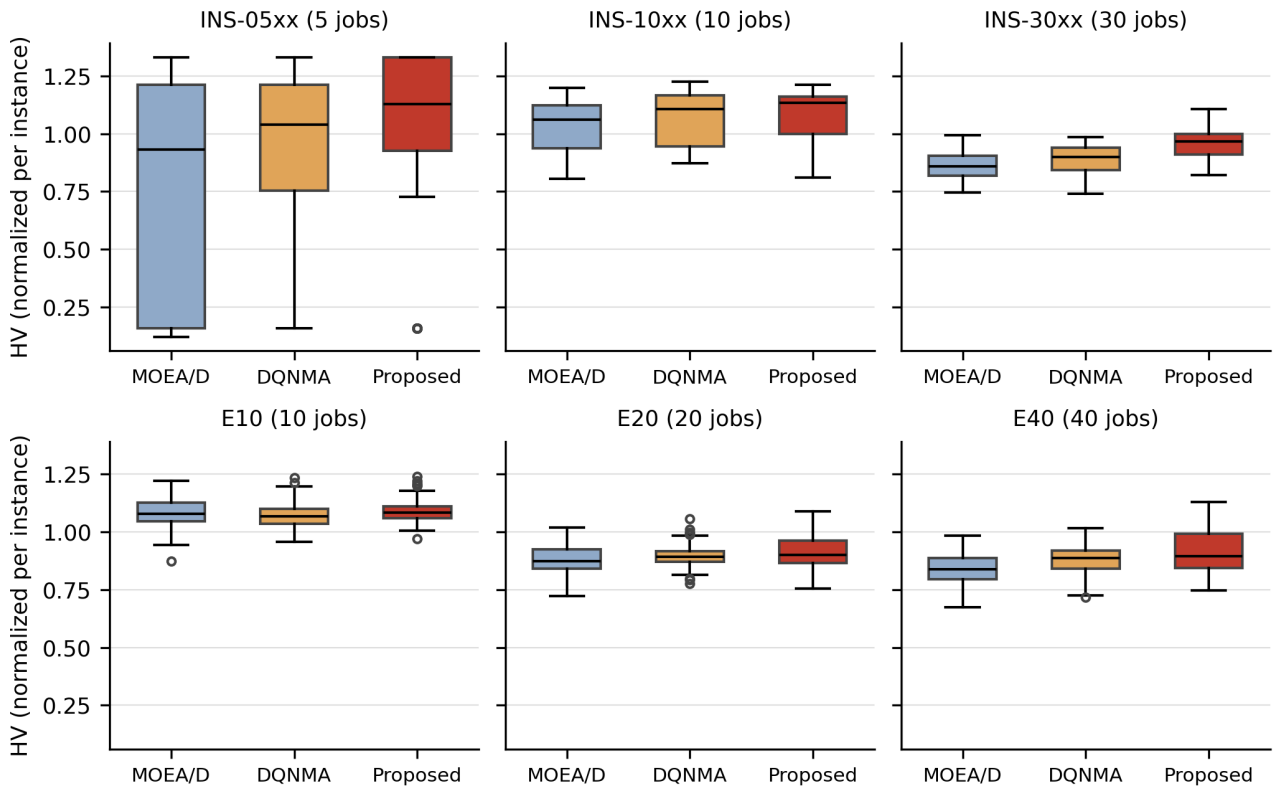


**Figure 7.** Distribution of *HV* over all runs of the compared algorithms, grouped by instance size

Averaged over the 15 case instances, the proposed method attains the highest *HV* (1.009), ahead of DQNMA (0.953) and MOEA/D (0.899). Per instance it is significantly better than MOEA/D on 9 and than DQNMA on 6 of the 15 instances and never significantly worse than either; across instances, the Wilcoxon signed-rank test on per-instance mean *HV* confirms both differences ( $p = 0.002$  and  $p = 0.005$ ). The advantage holds at every size, with mean *HV* of 0.978, 1.091 and 0.956 on the small, medium and large instances against 0.902, 1.065 and 0.891 for DQNMA and 0.797, 1.036 and 0.864 for MOEA/D. The margin over DQNMA is largest on the 30-job instances, where the proposed method wins on 4 of 5, which is consistent with the design: a path relinking walk transfers a large amount of structure between good schedules, which pays off on long encodings.

Per instance, however, most comparisons end in a statistical tie: the proposed method is indistinguishable from MOEA/D on 24 and from DQNMA on 28 of the 42 instances. A natural explanation is the sparsity of the attainable objective set, and we tested this explanation on all instances. For each instance the final non-dominated sets of the 30 runs of the three algorithms were pooled, and Table 10 reports the size of the pooled reference front, the share of runs that reach its best value in each objective, and the outcome of the comparisons. On the small instances the attainable set is indeed sparse: the reference front has 1 to 3 points and 79–100% of runs reach the best value of each objective, so the methods often return identical fronts and cannot be separated

by any test. The sensitivity study shows the same effect for parameter settings: on INS-0503 only 2 of the 12 alternative settings differ significantly from the default, against 12 on INS-1003 and 11 on INS-3003 (Section 4.5). Beyond the small instances, however, sparsity does not explain the ties. The reference fronts there contain 10 to 121 points and at most 6% of runs reach the best makespan, yet ties remain frequent, and across the instances the effect size of the difference between the proposed method and either baseline is unrelated to the size of the front (Spearman  $\rho = 0.03$ ,  $p = 0.86$  against MOEA/D;  $\rho = -0.03$ ,  $p = 0.83$  against DQNMA). Sparsity therefore accounts for the ties on the small instances only; on the larger ones the ties reflect genuinely close performance under the common budget rather than a lack of discriminating power.

**Table 10.** Density of the attainable objective set and separation of the algorithms by instance group

| Group    | Inst. | Front size | Runs reaching the best value (%) |      |        | Proposed method    |                   |
|----------|-------|------------|----------------------------------|------|--------|--------------------|-------------------|
|          |       |            | $C_{\max}$                       | $DT$ | $Cost$ | vs MOEA/D<br>W/T/L | vs DQNMA<br>W/T/L |
| INS-05xx | 5     | 2.0 (1–3)  | 79.3                             | 97.3 | 100.0  | 3/2/0              | 1/4/0             |

|          |   |               |     |      |      |       |       |
|----------|---|---------------|-----|------|------|-------|-------|
| INS-10xx | 5 | 16.4 (10–23)  | 5.3 | 61.3 | 96.7 | 3/2/0 | 1/4/0 |
| INS-30xx | 5 | 66.4 (54–89)  | 5.3 | 3.3  | 3.3  | 3/2/0 | 4/1/0 |
| E10      | 9 | 50.0 (30–71)  | 5.6 | 3.7  | 85.2 | 0/9/0 | 2/7/0 |
| E20      | 9 | 80.9 (62–121) | 3.7 | 3.3  | 7.0  | 3/6/0 | 2/7/0 |
| E40      | 9 | 69.2 (38–88)  | 4.1 | 3.3  | 3.3  | 6/3/0 | 3/5/1 |

Table 11 reports the best value attained on each objective for one representative instance of each size, separating objective-wise strengths from aggregate  $HV$ .

Table 11. Best objective values attained by each algorithm on representative instances

| Instance | Objective  | MOEA/D | DQNMA | Proposed |
|----------|------------|--------|-------|----------|
| INS-0503 | $C_{\max}$ | 79.5   | 79.5  | 78.8     |
|          | $DT$       | 17.0   | 17.0  | 17.0     |
|          | $Cost$     | 70.0   | 70.0  | 70.0     |
| INS-1003 | $C_{\max}$ | 89.0   | 87.6  | 87.4     |
|          | $DT$       | 268.8  | 277.1 | 268.4    |
|          | $Cost$     | 143.4  | 142.0 | 142.0    |
| INS-3003 | $C_{\max}$ | 170.4  | 166.5 | 162.9    |
|          | $DT$       | 699.1  | 745.6 | 682.6    |
|          | $Cost$     | 466.7  | 445.9 | 458.5    |

When the number of times each algorithm attains the best mean value of an objective is counted across the 15 case instances, with ties counted for each algorithm, the three methods show distinct profiles: MOEA/D attains the best tardiness on 12 instances, DQNMA the best transportation cost on 15, and the proposed method and DQNMA the best makespan on 9 each. No method is best on every objective, so the higher  $HV$  of the proposed method reflects a better balance across the three objectives rather than dominance in any one of them.

On the 27 additional instances (Table 8) the proposed method is significantly better than MOEA/D on 9 and then DQNMA on 7 instances, indistinguishable from them on 18 and 19 instances, respectively, and significantly worse only against DQNMA on E40-F1-T3. The signed-rank test on per-instance mean  $HV$  favors it over both MOEA/D ( $p = 0.003$ ) and DQNMA ( $p = 0.001$ ); over all 42 instances both differences are significant ( $p < 0.001$  in both cases). The objective-wise profile is sharper than on the case instances: MOEA/D attains the best mean tardiness on 25 of the 27 additional instances and DQNMA the best mean transportation cost on 19, while the proposed method attains the best makespan on 15 (DQNMA on 13).

#### 4.5 Parameter sensitivity

The sensitivity of the proposed method to its parameters was examined on one instance of each size, INS-0503, INS-1003 and INS-3003. With the default setting of Table 6 as the starting point, one parameter was changed at a time with all others kept at their default values, which gives 12 alternative settings (Table 12). The default VND visits  $N_1$ ,  $N_2$  and  $N_3$  in a fixed cyclic order; three alternatives replace this order by random selection of the next neighborhood with probabilities  $(1/3, 1/3, 1/3)$ ,  $(0.5, 0.25, 0.25)$  and  $(0.25, 0.25, 0.5)$  for  $(N_1, N_2, N_3)$ . The restart probability  $p_{\text{restart}}$  was set to 0.05, 0.2 and 0.3, the path relinking sample size per step to 3, 9 and 12, and the path relinking budget per call to 20, 50 and 70. The method has no separate guidance-rate parameter: how far a relinking walk proceeds towards the guiding solution is governed by the budget per call, how greedily each step is chosen by the sample size, and how often relinking is used at all by  $1 - p_{\text{restart}}$ , so these three parameters jointly represent path relinking. Each setting was run 10 times per instance.  $HV$  is normalized over the union of the solutions of all 13 settings on each instance with the reference point  $(1.1, 1.1, 1.1)$ , and each alternative is compared with the default by the Wilcoxon rank-sum test at the 0.05 level.

The default setting attains the highest mean  $HV$  over the three instances (1.140) and is the best of the 13 settings on INS-1003 and INS-3003, where 23 of the 24 alternatives are significantly worse. On INS-0503 the setting hardly matters apart from the two largest relinking budgets, all alternatives lie within 0.03 of the default and none differs from it significantly, because on this instance 99.2% of runs reach the same best tardiness and every run the same best transportation cost. The relinking budget per call is the most influential parameter. Raising it to 50 or 70 decodings lowers the mean  $HV$  to 0.925 and 0.827, respectively, because under a fixed evaluation budget each longer walk leaves fewer iterations for restarts and local search, whereas lowering it to 20 is harmless on the small instance but significantly worse on the two larger ones, where a short walk transfers too little structure. The restart probability behaves in the same way: fewer restarts (0.05) and more restarts (0.2 and 0.3) are both worse on the medium and large instances, so relinking without diversification and diversification without relinking are each inferior to their combination. A sample size of 6 per step is likewise better than 3, which leaves each step of the walk nearly blind, and then 9 or 12, which spend the relinking budget on evaluating candidates rather than on advancing along the path. Replacing the fixed neighborhood order by random selection does not help selection biased towards  $N_1$  or  $N_3$  is clearly detrimental on the larger instances (mean  $HV$  1.024 and 1.013), and uniform selection comes closest to the fixed order (1.105) but is still significantly worse on INS-1003. On the medium and large instances no single-parameter change improves on the default, and for the three numerical parameters the default lies between a worse lower and a worse upper value; the conclusions of Sections 4.3 and 4.4 therefore do not rest on a fortunate parameter choice.

Table 12. Sensitivity of the proposed method to one-at-a-time parameter changes

| Parameter                                               | Value             | INS-0503        | INS-1003 | INS-3003        | Mean $HV$ |
|---------------------------------------------------------|-------------------|-----------------|----------|-----------------|-----------|
| Default setting (Table 6)                               | —                 | 1.253           | 1.136    | 1.032           | 1.140     |
| VND neighborhood selection: fixed                       | (1/3, 1/3, 1/3)   | 1.276 $\approx$ | 1.064 –  | 0.976 $\approx$ | 1.105     |
| cyclic order by default; alternatives                   | (0.5, 0.25, 0.25) | 1.239 $\approx$ | 0.962 –  | 0.872 –         | 1.024     |
| select at random with probabilities                     | (0.25, 0.25, 0.5) | 1.233 $\approx$ | 0.941 –  | 0.865 –         | 1.013     |
| Restart probability $p_{\text{restart}}$ , default 0.10 | 0.05              | 1.226 $\approx$ | 1.007 –  | 0.905 –         | 1.046     |
|                                                         | 0.2               | 1.279 $\approx$ | 1.007 –  | 0.925 –         | 1.071     |
|                                                         | 0.3               | 1.254 $\approx$ | 0.974 –  | 0.929 –         | 1.052     |
| Path relinking sample size per step, default 6          | 3                 | 1.263 $\approx$ | 1.019 –  | 0.904 –         | 1.062     |
|                                                         | 9                 | 1.272 $\approx$ | 0.967 –  | 0.909 –         | 1.050     |
|                                                         | 12                | 1.269 $\approx$ | 1.020 –  | 0.927 –         | 1.072     |
| Path relinking budget per call, default 35              | 20                | 1.260 $\approx$ | 1.024 –  | 0.926 –         | 1.070     |
|                                                         | 50                | 1.150 –         | 0.877 –  | 0.749 –         | 0.925     |
|                                                         | 70                | 1.044 –         | 0.776 –  | 0.659 –         | 0.827     |

## 5. Conclusion

This paper addresses collaborative production and delivery scheduling in distributed processing and assembly factories. The DFAJSP-T model is formulated to minimize makespan, total tardiness and transportation cost simultaneously, and a decomposition-based GRASP with path relinking is developed, in which information transfer between good schedules is made explicit and local search is guided by the critical path.

The experiments on 42 instances support three conclusions. First, distributed collaboration is primarily a logistics lever: relative to independent production, collaborative production with transportation constraints reduces transportation cost on average by 11.6% on the case instances and by 11.8% on the additional ones while changing the makespan by less than 1%, because only the assembly stage is cross-factory flexible and the processing stage remains the bottleneck. Second, cross-factory assembly reduces rather than increases transportation cost, because fixing the assembly workshop by customer proximity optimizes the delivery arc while ignoring the frequently more expensive processing-to-assembly arc; consistent with this mechanism, the saving rises from 10.7% to 22.6% when inter-factory unit costs are tripled and falls to 2.0% when delivery costs are tripled, and the share of jobs assembled away from their nearest factory is interior, between 0.15 and 0.35 depending on the instance group. Third, larger solution spaces are not better under a fixed evaluation budget: unrestricted collaboration is never better than the constrained strategy and is worse on 36 of the 42 instances, and on the largest order books (30 and 40 jobs) independent production attains a significantly higher hypervolume than collaborative production on 12 of 14 instances, with better makespan and tardiness, while collaboration keeps a smaller cost advantage. There, the relocations that save transportation cost concentrate assembly on the site co-located with the processing workshop, whose assembly lines approach saturation, so the saving is paid for in queueing.

On the algorithmic side, the proposed method attains a significantly higher hypervolume than both MOEA/D and DQNMA across the instances (signed-rank  $p < 0.001$  against both), is never significantly worse than MOEA/D and loses

to DQNMA on one instance. Per instance, however, most comparisons are statistical ties (24 of 42 against MOEA/D and 28 against DQNMA), and the methods have distinct objective-wise profiles: MOEA/D is strongest on tardiness, DQNMA on transportation cost, and the proposed method balances the three objectives best. The sparsity of the attainable objective set explains the exact ties on the smallest instances, whose reference fronts contain one to three points, but across the instances the size of the front is unrelated to the separation between the methods, so the ties on larger instances reflect close performance rather than a lack of discriminating power. The sensitivity analysis shows that the default parameters are robust, the path relinking budget per call being the most influential parameter.

The instances are derived from a single firm with one processing stage and at most four sites, and the evaluation budget was fixed rather than scaled with the number of operations. Future work will extend the model to dynamic scheduling, since real production is subject to order insertion, machine breakdown and delayed transportation, and will study other network topologies, such as several processing sites, multi-level assembly and hub-based transportation, together with budgets that grow with instance size, under which the relative merit of collaborative and independent production on large order books may change.

## References

- [1] Duran, E., Ozturk, C., O'Sullivan, B. Planning and Scheduling Shared Manufacturing Systems: Key Characteristics, Current Developments and Future Trends. *Int. J. Prod. Res.* 2025;63(13):4958–4990. DOI: 10.1080/00207543.2024.2442549
- [2] Kazemi, H., Noureldath, M., Fathollahi-Fard, A.M., Gendreau, M. Integrated Two-Stage Multi-Factory Assembly Scheduling with Maintenance Considerations. *Int. J. Prod. Res.* 2025;63(6):2133–2149. DOI: 10.1080/00207543.2024.2396512
- [3] Dautère-Pères, S., Ding, J., Shen, L., Tamssaouet, K. The Flexible Job Shop Scheduling Problem: A Review. *Eur. J. Oper. Res.* 2024;314(2):409–432. DOI: 10.1016/j.ejor.2023.05.017
- [4] Özgüven, C., Özbakır, L., Yavuz, Y. Mathematical Models for Job-Shop Scheduling Problems with Routing and Process Plan

- Flexibility. *Appl. Math. Model.* 2010;34(6):1539–1548. DOI: 10.1016/j.apm.2009.09.002
- [5] Zhang, S., Wang, S. Flexible Assembly Job-Shop Scheduling with Sequence-Dependent Setup Times and Part Sharing in a Dynamic Environment: Constraint Programming Model, Mixed-Integer Programming Model, and Dispatching Rules. *IEEE Trans. Eng. Manag.* 2018;65(3):487–504. DOI: 10.1109/TEM.2017.2785774
- [6] Wu, X., Liu, X., Zhao, N. An Improved Differential Evolution Algorithm for Solving a Distributed Assembly Flexible Job Shop Scheduling Problem. *Memetic Comput.* 2019;11(4):335–355. DOI: 10.1007/s12293-018-00278-7
- [7] Hu, Y., Zhang, L., Wang, Q., Zhang, Z., Tang, Q. A Matheuristic-Based Multi-Objective Evolutionary Algorithm for Flexible Assembly Job Shop Scheduling Problem in Cellular Manufacture. *Swarm Evol. Comput.* 2024;87:101549. DOI: 10.1016/j.swevo.2024.101549
- [8] Hu, Y., Zhang, L., Zhang, Z., Li, Z., Tang, Q. Matheuristic and Learning-Oriented Multi-Objective Artificial Bee Colony Algorithm for Energy-Aware Flexible Assembly Job Shop Scheduling Problem. *Eng. Appl. Artif. Intell.* 2024;133:108634. DOI: 10.1016/j.engappai.2024.108634
- [9] Naderi, B., Ruiz, R. The Distributed Permutation Flowshop Scheduling Problem. *Comput. Oper. Res.* 2010;37(4):754–768. DOI: 10.1016/j.cor.2009.06.019
- [10] Hatami, S., Ruiz, R., Andrés-Romano, C. The Distributed Assembly Permutation Flowshop Scheduling Problem. *Int. J. Prod. Res.* 2013;51(17):5292–5308. DOI: 10.1080/00207543.2013.807955
- [11] De Giovanni, L., Pezzella, F. An Improved Genetic Algorithm for the Distributed and Flexible Job-Shop Scheduling Problem. *Eur. J. Oper. Res.* 2010;200(2):395–408. DOI: 10.1016/j.ejor.2009.01.008
- [12] Meng, L., Zhang, C., Ren, Y., Zhang, B., Lv, C. Mixed-Integer Linear Programming and Constraint Programming Formulations for Solving Distributed Flexible Job Shop Scheduling Problem. *Comput. Ind. Eng.* 2020;142:106347. DOI: 10.1016/j.cie.2020.106347
- [13] Behnamian, J., Fatemi Ghomi, S.M.T. A Survey of Multi-Factory Scheduling. *J. Intell. Manuf.* 2016;27(1):231–249. DOI: 10.1007/s10845-014-0890-y
- [14] Lohmer, J., Lasch, R. Production Planning and Scheduling in Multi-Factory Production Networks: A Systematic Literature Review. *Int. J. Prod. Res.* 2021;59(7):2028–2054. DOI: 10.1080/00207543.2020.1797207
- [15] Zhang, Q., Manier, H., Manier, M.-A. A Genetic Algorithm with Tabu Search Procedure for Flexible Job Shop Scheduling with Transportation Constraints and Bounded Processing Times. *Comput. Oper. Res.* 2012;39(7):1713–1723. DOI: 10.1016/j.cor.2011.10.007
- [16] Chan, F.T.S., Prakash, A., Ma, H.L., Wong, C.S. A Hybrid Tabu Sample-Sort Simulated Annealing Approach for Solving Distributed Scheduling Problem. *Int. J. Prod. Res.* 2013;51(9):2602–2619. DOI: 10.1080/00207543.2012.737948
- [17] Luo, Q., Deng, Q., Gong, G., Zhang, L., Han, W., Li, K. An Efficient Memetic Algorithm for Distributed Flexible Job Shop Scheduling Problem with Transfers. *Expert Syst. Appl.* 2020;160:113721. DOI: 10.1016/j.eswa.2020.113721
- [18] Du, Y., Li, J., Luo, C., Meng, L. A Hybrid Estimation of Distribution Algorithm for Distributed Flexible Job Shop Scheduling with Crane Transportations. *Swarm Evol. Comput.* 2021;62:100861. DOI: 10.1016/j.swevo.2021.100861
- [19] Homayouni, S.M., Fontes, D.B., Gonçalves, J.F. A Multistart Biased Random Key Genetic Algorithm for the Flexible Job Shop Scheduling Problem with Transportation. *Int. Trans. Oper. Res.* 2023;30(2):688–716. DOI: 10.1111/itor.12878
- [20] Du, B., Han, S., Guo, J., Li, Y. A Hybrid Estimation of Distribution Algorithm for Solving Assembly Flexible Job Shop Scheduling in a Distributed Environment. *Eng. Appl. Artif. Intell.* 2024;133:108491. DOI: 10.1016/j.engappai.2024.108491
- [21] Chen, Z.-L. Integrated Production and Outbound Distribution Scheduling: Review and Extensions. *Oper. Res.* 2010;58(1):130–148. DOI: 10.1287/opre.1080.0688
- [22] Archetti, C., Peirano, L., Speranza, M.G. Optimization in Multimodal Freight Transportation Problems: A Survey. *Eur. J. Oper. Res.* 2022;299(1):1–20. DOI: 10.1016/j.ejor.2021.07.031
- [23] González, M.A., Vela, C.R., Varela, R. Scatter Search with Path Relinking for the Flexible Job Shop Scheduling Problem. *Eur. J. Oper. Res.* 2015;245(1):35–45. DOI: 10.1016/j.ejor.2015.02.052
- [24] Martí, R., Campos, V., Resende, M.G.C., Duarte, A. Multiobjective GRASP with Path Relinking. *Eur. J. Oper. Res.* 2015;240(1):54–71. DOI: 10.1016/j.ejor.2014.06.042
- [25] Zhang, H., Chen, Y., Xu, G., Zhang, Y. Distributed Assembly Flexible Job Shop Scheduling with Dual-Resource Constraints via a Deep Q-Network Based Memetic Algorithm. *Swarm Evol. Comput.* 2025;98:102086. DOI: 10.1016/j.swevo.2025.102086
- [26] Feo, T.A., Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. *J. Glob. Optim.* 1995;6(2):109–133. DOI: 10.1007/BF01096763
- [27] Glover, F. Tabu Search and Adaptive Memory Programming — Advances, Applications and Challenges. In: *Interfaces in Computer Science and Operations Research*, pp. 1–75. Kluwer, 1997. DOI: 10.1007/978-1-4615-4102-8\_1
- [28] Laguna, M., Martí, R. GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. *INFORMS J. Comput.* 1999;11(1):44–52. DOI: 10.1287/ijoc.11.1.44
- [29] Mladenović, N., Hansen, P. Variable Neighborhood Search. *Comput. Oper. Res.* 1997;24(11):1097–1100. DOI: 10.1016/S0305-0548(97)00031-2
- [30] Das, I., Dennis, J.E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. Optim.* 1998;8(3):631–657. DOI: 10.1137/S1052623496307510
- [31] Zhang, Q., Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* 2007;11(6):712–731. DOI: 10.1109/TEVC.2007.892759
- [32] Nowicki, E., Smutnicki, C. A Fast Taboo Search Algorithm for the Job Shop Problem. *Manag. Sci.* 1996;42(6):797–813. DOI: 10.1287/mnsc.42.6.797
- [33] Mastrolilli, M., Gambardella, L.M. Effective Neighbourhood Functions for the Flexible Job Shop Problem. *J. Sched.* 2000;3(1):3–20. DOI: 10.1002/(SICI)1099-1425(200001/02)3:1<3::AID-JOS32>3.0.CO;2-Y
- [34] Zitzler, E., Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. Evol. Comput.* 1999;3(4):257–271. DOI: 10.1109/4235.797969