Simultaneous Scheduling of Processing Machines and Automated Guided Vehicles via a Multi-View Modeling-Based Hybrid Algorithm

Bin Xin, *Member, IEEE*, Sai Lu, Qing Wang, Fang Deng, *Senior Member, IEEE*, Xiang Shi, Jun Cheng, *Member, IEEE*, and Yuhang Kang

***Abstract*— The flexible job-shop co-scheduling problem (FJCSP) for processing machines and automated guided vehicles (AGVs) in a flexible manufacturing system (FMS) has attracted more attention with the aim of improving production efficiency. In FMS, AGVs in charge of transporting jobs realize the flexible linkage of operations between different processing machines. The added interdependence between transporting and processing tasks brings more difficulties than the traditional flexible job-shop scheduling problem (FJSP). In this paper, the mathematical model of FJCSP is formulated to minimize the makespan. Considering the feature similarity of FJCSP with FJSP and AGV-routing problem in different cases, a multi-view modeling-based hybrid algorithm consisting of an estimation of distribution algorithm (EDA) and an ant colony optimization (ACO) is proposed. In EDA, a probability model abstracts the information in superior solutions about the operation sequencing and the rule selection for scheduling machines and AGVs. In ACO, a job-path pheromone model and an AGV-path pheromone model are designed to jointly select the job-machine-AGV combination with shorter processing time and transportation time. In the proposed hybrid algorithm, EDA and ACO generate solutions independently and achieve cooperation by sharing elites. An adaptive parameter is designed to regulate the use of the two methods to adapt to the varying demands of multi-view modeling in different cases and search stages. Furthermore, a local search with a three-layer operator based on the critical path method is proposed to balance exploration and exploitation in solution space. Finally, computational experiments involving a case study verified the advantage of the multi-view modeling-based hybrid algorithm in comparison with the state-of-the-art approaches.**

***Note to Practitioners*—This paper was motivated by the optimization problem of scheduling machines and automated guided vehicles (AGVs) in flexible manufacturing system (FMS). In FMS with AGVs, the transportation stages for jobs by AGVs significantly impact the overall production efficiency of the FMS and cannot be overlooked. This paper suggested a hybrid evolutionary algorithm using an estimation of distribution algorithm (EDA), an ant colony optimization (ACO) and a local search algorithm based on the critical path method. In the proposed hybrid algorithm, an adaptive parameter is introduced to regulate the utilization of EDA and ACO in generating a new population. This paper presents a mathematical characterization of the scheduling problem and subsequently outlines the step-by-step design of the hybrid algorithm. Computational experiments, including a case study, demonstrate that the hybrid algorithm exhibits adaptability to various instances and outperforms state-of-the-art approaches.**

***Index Terms*— Simultaneous scheduling, Flexible manufacturing system, Automated guided vehicle, Modeling-based method, Multi-view.**

I. Introduction

WITH the continuous development of user needs and the flexibility of manufacturing, the manufacturing mode has gradually changed from the product-driven mode to the user-driven mode. In order to meet these demands, the flexible manufacturing system (FMS) has become an important development direction of the industry in the future because of its high automation, fast response speed and low-cost line change. The flexible job-shop scheduling problem (FJSP) is always a research hotspot that has been proven to be an NP-hard problem [1], [2]. With the development of production automation and information integration, modern industrial production puts forward higher requirements for logistics systems. As a kind of automatic equipment, automated guided vehicle (AGV) can complete specified transportation tasks in a given path or environment layout. AGVs accomplish the flexible

Manuscript received 26 May 2023; accepted 23 July 2023. This article was recommended for publication by Associate Editor Z. Pei and Editor T. Nishi upon evaluation of the reviewers' comments. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1308000, in part by the National Outstanding Youth Talents Support Program under Grant 61822304, in part by the Basic Science Center Programs of NSFC under Grant 62088101, in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, in part by the Shanghai Municipal Commission of Science and Technology Project under Grant 19511132101, in part by the National Natural Science Fund of China under Grant 62003044, and in part by the National Science Fund for Distinguished Young Scholars of China under Grant 62025301. *(Corresponding author: Bin Xin.)*

Bin Xin, Sai Lu, Qing Wang, and Xiang Shi are with the National Key Laboratory of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: brucebin@bit.edu.cn; 292243532@qq.com; wangqing1020@bit.edu.cn; bit_shixiang@163.com).

Fang Deng is with the National Key Laboratory of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China (e-mail: dengfang@bit.edu.cn).

Jun Cheng and Yuhang Kang are with the Key Laboratory of Human–Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: jun.cheng@siat.ac.cn; yh.kang1@siat.ac.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TASE.2023.3301656.

Digital Object Identifier 10.1109/TASE.2023.3301656

operation linkages between various processing machines and play an important role in the logistics system of FMS. This paper focuses on the flexible job-shop co-scheduling problem (FJCSP) for machines and AGVs to improve system flexibility and production efficiency. From a practical perspective, FJCSP is widely present in industries such as intelligent manufacturing of hardware products [3] and automated container terminals [4]. From a theoretical perspective, although FJCSP can be regarded as a variant of FJSP, it becomes more complex due to the increase in scheduling entities. Therefore, FJCSP is a practically urgent and theoretically valuable optimization problem.

In FJCSP, the completion time of jobs consists of processing time and no-load/load time, which is affected by the processing sequence on machines and the transportation sequence of AGVs together. The sequential-logic constraint between the transportation stage and processing stage of each operation makes it difficult to decouple the problem. The difficulty of FJCSP increases obviously in contrast with FJSP, which is mainly reflected in the following aspects:

1) In addition to the scheduling problem of processing machines under the operation constraints, the task allocation problem of the AGVs also needs to be solved, which will commonly expand the solution space;
2) The transport behaviors of AGVs including no-load and load phases are significantly different from the processing behaviors of machines. These two types of behaviors are executed alternately, and necessary waiting phases may occur during execution. This brings a more complex cascading effect for calculating the completion time of jobs;
3) The interdependence between processing tasks and transportation tasks is difficult to decouple, which means layering or decoupling mechanisms are likely to lose the optimal solution.

The practical urgency of simultaneous scheduling in FMS with AGVs and the difficulty have attracted some scholars to study this problem. Recently, Li conducted a review on integrated flexible job shop scheduling problems, in which the co-scheduling problem with AGV was mentioned [5]. The early research on FJCSP dates back to the 1990s. Doo et al. set two AGV models based on Petri nets for the material handling and a model for the processing stage. These models were integrated into a single coherent model and a scheduling method that employed Petri nets, and a heuristic search was used for a better solution [6]. Bilge and Ulusoy established a nonlinear mixed-integer programming model and proposed a heuristic algorithm to generate machine schedules and a sliding time windows heuristic for the vehicle scheduling [7]. Later, more and more scholars began to pay attention to this problem. Lacomme et al. researched the simultaneous job input sequencing and vehicle dispatching problem in FMS with a single AGV [8]. Then, they proposed a branch-and-bound coupled with a discrete event simulation model to evaluate the job sequence under the given vehicle and machine dispatching rules [9].

As a popular evolutionary algorithm, genetic algorithm (GA) has been designed into many variants to solve similar problems. Ulusoy et al. adopted a genetic algorithm with a special uniform crossover operator and two mutation operators, and lots of instances were solved to evaluate the proposed algorithm [10]. Chaudhry et al. proposed a spreadsheet-based GA approach to solving the co-scheduling problem. An adaptation of the proprietary GA software was demonstrated to minimize the total completion time or makespan in FMS [11]. Badakhshian et al. adopted a fuzzy controller to control the behavior of the GA during solving the scheduling problem [12]. Nouri et al. proposed a hybrid metaheuristic approach based on clustered holonic multiagent model. A neighborhood-based GA for global exploration was adopted and a set of cluster agents used tabu search for promising regions [13]. Meng et al. designed a diversity check method into GA to prevent the decline of population diversity [14].

Some other evolutionary algorithms were also employed to solve FJCSP. Gnanavel Babu et al. proposed a new meta-heuristic differential evolution algorithm to minimize the makespan [15]. Tatsushi et al. addressed a bilevel decomposition algorithm for solving the simultaneous scheduling and conflict-free routing problems for AGVs to minimize the total weighted tardiness. A mixed-integer formulation was decomposed into task assignment and routing subproblems which were solved by using Lagrangian relaxation [16]. Zheng et al. established a mixed-integer linear programming model and proposed heuristic algorithm based on tabu search, which included a novel two-dimensional solution representation and the generating operation of two neighbor solutions [17]. Wen et al. designed a harmony search algorithm with a neighborhood search incorporating problem knowledge to guide the search [18].

More hybrid or multi-stage algorithms were proposed to solve the complex problem. Saidi Mehrabad et al. considered the scheduling problem containing the conflict-free routing problem for AGVs and the basic job-shop scheduling problem, and proposed a two-stage ant colony algorithm to solve it [19]. Considering real situations such as deadlock or blockage, Lin et al. proposed a hybrid algorithm based on GA and optimal computing budget allocation to reduce simulation replications [20]. Nageswararao et al. proposed a new meta-heuristic gravitational search algorithm to solve the simultaneous scheduling problem and the adequacy of the algorithm was demonstrated by comparing it with existing results [21]. Lyu et al. adopted a hybrid consisting of GA and Dijkstra algorithm to solve the scheduling problem with the objectives of minimizing the makespan and the transportation time [22].

Despite the above research, the problem is still worth studying further. As most scholars have stated, FJCSP is an expansion of the traditional FJSP [23]. However, on the other hand, due to the AGVs introduced into the problem, the routing issue of AGVs arises and may even become dominant in some cases. Therefore, the approach to solving this problem needs to be designed from the above two views. As we know, the estimation of distribution algorithm (EDA) is a novel evolutionary algorithm based on statistical learning theory and has been applied widely to similar scheduling problems [2].

EDA estimates the probability distribution of elite individuals and builds a probability model of the most promising area by statistical information according to the historical search experience. In addition, the ant colony algorithm (ACO) has been widely applied to path-related scheduling problems such as the path planning problem [24] and vehicle routing problem [25], [26]. EDA and ACO are both based on statistical models, namely the **probability model** and the **pheromone model**, respectively. Compared with the other methods like GAs, EDA and ACO are conducive to absorbing various problem knowledge. For a complex problem, reasonable heuristics designed by fully using the problem knowledge can commonly achieve better performance.

This paper intends to use the two modeling-based methods to extract and utilize the statistical information of the elite solutions for global exploration through their complementation and collaboration from different views. The main innovations and contributions of this paper include the following issues:

1) To the best of our knowledge, this is the first attempt to solve FJCSP by the cooperation of two modeling-based methods described different features of elites, i.e., EDA and ACO. The two methods generate new solutions independently and share elites to update respective models. The use of them is adjusted adaptively to adapt to the varying demands of multi-view modeling in different cases or different search stages;
2) Inspired by successful application of EDA to the traditional FJSP, a probability model in EDA is designed to abstract the information in superior solutions about the operation positions and the rule selection for scheduling machines and AGVs is designed for generating a solution;
3) Considering the AGVs introduced into the problem and the path issue arising, a job-path pheromone model and an AGV-path pheromone model in ACO are designed to jointly determine the selection of the job-machine-AGV combinations to generate a solution;
4) In view of the three types of path segments in the critical path, a local search with a targeted three-layer operator is designed to improve elites. The hybrid algorithm combining two modeling-based methods for global exploration and the local search for local exploitation shows advantages in different cases.

The remainder of this paper is organized as follows. The problem model and the objective function are described in Section II and the details about the proposed hybrid algorithm are displayed in Section III. Finally, the experiment results are shown and analyzed to verify the proposed algorithm in Section IV and the conclusions are presented in Section V.

II. Problem Formulation

A. Problem Description

In an FMS with AGVs as illustrated in Fig.1, there are three kinds of entities to be considered: jobs, machines and AGVs. Each job contains a series of operations to be processed sequentially by different machines. Each machine has the ability to process more than one operation, and each AGV can carry different jobs between any two machines for the next operations along the planned or given path. In particular, the different operations of a job can be transported by different AGVs. FJCSP can be divided into four sub-problems:

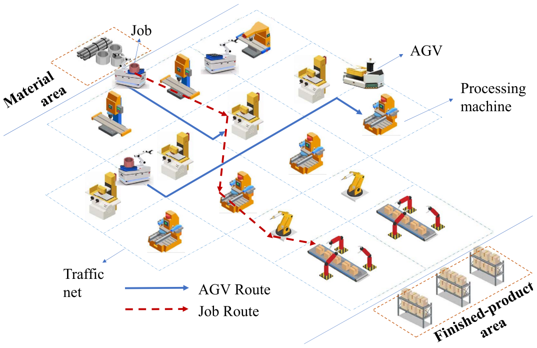


Fig. 1. Flexible manufacturing system with AGVs.

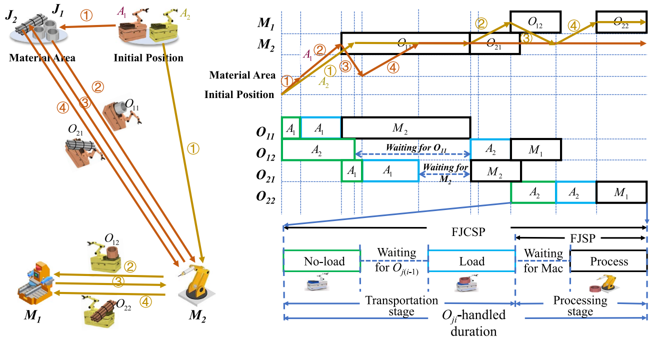


Fig. 2. A simple handling procedure of two jobs.

1) *The assignment problem of processing tasks*: assign a processing machine for each operation;
2) *The assignment problem for transportation tasks*: assign a transport AGV for each operation;
3) *The sequencing problem of processing tasks*: plan a processing sequence for each machine;
4) *The sequencing problem of transportation tasks*: plan a transporting sequence for each AGV.

In this paper, the objective of the problem is to minimize the makespan, i.e., the time when all the jobs have been processed. The notations are listed in Table I. The decision variables are X_{jim}, $Y^m_{j_1i_1,j_2i_2}$, Z_{jik} and $W^k_{j_1i_1,j_2i_2}$, corresponding to the four sub-problems aforementioned.

The handling procedure for two jobs is shown in Fig. 2, where ①-④ represents the routing sequence of AGVs. A complete operation-handled duration contains the no-load moving stage of the AGV, the load transportation stage for the job and the processing stage on the machine. For O_{22}-handled duration in Fig. 2, AGV A_2 will move from the machine M_1 to M_2 to take job J_2 (no-load stage). Then, it will deliver J_2 from M_2 to the target machine M_1 for processing O_{22} (load stage). The processing stage of O_{22} on M_1 is the same as in FJSP (processing stage). The interaction phase between different entities, e.g. the phase when jobs are taken from machines to AGVs, will cause some waiting time, such as the waiting time for O_{11} and the waiting time for M_2.

Remark 1: From Fig. 2, it can be found that the added no-load/load stages and more waiting stages in FJCSP in

TABLE I

DESCRIPTION OF KEY SYMBOLS

Symbol	Description
N_J	The number of jobs to be processed.
N_M	The number of machines in FMS.
N_A	The number of AGVs in FMS.
J	The index set of jobs, $J = \{1, 2, ..., N_J\}$.
J_j	The j-th job, $j \in J$.
M	The index set of machines, $M = \{1, 2, ..., N_M\}$.
M_m	The m-th machine, $m \in M$.
A	The index set of AGVs, $A = \{1, 2, ..., N_A\}$.
A_k	The k-th AGV, $k \in A$.
N_j^{opr}	The total number of the operations of J_j, $j \in J$.
N_O	The total number of all the operations, $N_O = \sum_{j=1}^{N_J} N_j^{opr}$.
O_j^{opr}	The index set of the operations of J_j, $O_j^{opr} = \{1, 2, ..., N_j^{opr}\}$, $j \in J$.
O_{ji}	The i-th operation of J_j, $i \in O_j^{opr}$.
S_{ji}^P	The processing stage of O_{ji}.
S_{ji}^T	The loaded transportation stage of O_{ji}.
$S_{ji}^{T'}$	The no-load moving stage of O_{ji}.
Ω_{ji}	The index set of machines which can handle O_{ji}.
t_{jim}^P	The processing time of O_{ji} on M_m, $m \in \Omega_{ji}$.
D	The position index set of the raw material area and machines, $D = \{0, 1, ..., N_M\}$.
D_d	The position set of the raw material area and machines, $d \in D$, D_0 means the position of raw material and M_m is located at D_m, $m \in M$.
$t'_{d_1 d_2}$	The no-load time between two positions, $\forall d_1, d_2 \in D$.
$t_{d_1 d_2}$	The transportation time between two positions, $\forall d_1, d_2 \in D$.
Tst'_{ji}	The starting time of the no-load stage $S_{ji}^{T'}$ for O_{ji}.
Tet'_{ji}	The ending time of the no-load stage $S_{ji}^{T'}$ for O_{ji}.
Tst_{ji}	The starting time of the transportation stage S_{ji}^T for O_{ji}.
Tet_{ji}	The ending time of the transportation stage S_{ji}^T for O_{ji}.
Tsp_{ji}	The starting time of the processing stage S_{ji}^P of O_{ji}.
Tep_{ji}	The ending time of the processing stage S_{ji}^P of O_{ji}.
X_{jim}	If O_{ji} is processed on M_m, $X_{jim} = 1$; otherwise, $X_{jim} = 0$.
$Y_{j_1 i_1, j_2 i_2}^m$	If $O_{j_1 i_1}$ is processed on M_m before $O_{j_2 i_2}$, $Y_{j_1 i_1, j_2 i_2}^m = 1$; otherwise, $Y_{j_1 i_1, j_2 i_2}^m = 0$.
Z_{jik}	If A_k takes O_{ji}, $Z_{jik} = 1$; otherwise, $Z_{jik} = 0$.
$W_{j_1 i_1, j_2 i_2}^k$	If A_k takes $O_{j_1 i_1}$ before $O_{j_2 i_2}$, $W_{j_1 i_1, j_2 i_2}^k = 1$; otherwise, $W_{j_1 i_1, j_2 i_2}^k = 0$.

contrast with FJSP bring more obvious coupling relations, which increases the difficulty of solving this problem.

In order to grasp the problem backbone and simplify redundancy, some assumptions are necessary and are shown as follows:

1) Each machine has sufficient input/output buffer space for storing the jobs;
2) Each machine is allowed to process at most one operation at any time;
3) Each operation is not preemptive;
4) The processing time of each operation on different machines are available and deterministic;
5) All AGVs are identical and cost the same and predetermined transportation time on the same path;
6) Each AGV will start its first task at the material area;
7) Each AGV is allowed to transport at most one job at any time;
8) The traffic congestion and path conflict of AGVs are not taken into account;
9) When completing a transportation task, the AGV will move to the next transportation task and wait until the task can be executable.

B. Constraints

According to the above descriptions, the following constraints must be satisfied.

$$\sum_{m \in \Omega_{ji}} X_{jim} = 1, \quad \forall j \in J, \ \forall i \in O_j^{opr} \tag{1}$$

$$\sum_{k \in A} Z_{jik} = 1, \quad \forall j \in J, \ \forall i \in O_j^{opr} \tag{2}$$

$$\sum_{j_1=1}^{N_J} \sum_{i_1=1}^{N_{j_1}^{opr}} Y_{j_1 i_1, j_2 i_2}^m \cdot X_{j_2 i_2 m} \leq 1, \forall j_2 \in J, \quad \forall i_2 \in O_{j_2}^{opr}, \forall m \in M \tag{3}$$

$$\sum_{j_1=1}^{N_J} \sum_{i_1=1}^{N_{j_1}^{opr}} \sum_{j_2=1}^{N_J} \sum_{i_2=1}^{N_{j_2}^{opr}} Y_{j_1 i_1, j_2 i_2}^m \cdot X_{j_1 i_1 m} \cdot X_{j_2 i_2 m} = Q_m - 1, \ \forall m \in M \tag{4}$$

$$\sum_{j_1=1}^{N_J} \sum_{i_1=1}^{N_{j_1}^{opr}} W_{j_1 i_1, j_2 i_2}^k \cdot Z_{j_2 i_2 k} \leq 1, \forall j_2 \in J, \quad \forall i_2 \in O_{j_2}^{opr}, \ \forall k \in A \tag{5}$$

$$\sum_{j_1=1}^{N_J} \sum_{i_1=1}^{N_{j_1}^{opr}} \sum_{j_2=1}^{N_J} \sum_{i_2=1}^{N_{j_2}^{opr}} W_{j_1 i_1, j_2 i_2}^k \cdot Z_{j_1 i_1 k} \cdot Z_{j_2 i_2 k} = H_k - 1, \ \forall k \in A \tag{6}$$

where $Q_m = \sum_{j_1=1}^{N_J} \sum_{i_1=1}^{N_{j_1}^{opr}} X_{j_1 i_1 m}$ means the total number of processing tasks of the M_m ($m \in M$) and $H_k = \sum_{j_1=1}^{N_J} \sum_{i_1=1}^{N_{j_1}^{opr}} Z_{j_1 i_1 k}$ means the total number of transportation tasks of the A_k ($k \in A$).

Eq. (1) means that each operation can be processed by only one machine. Eq. (2) means that each operation can be transported by only one AGV. Eqs. (3)-(4) imply that each machine will handle all of its assigned operations sequentially. Eqs. (5)-(6) imply that each AGV will perform its transportation tasks sequentially.

C. Objective

The optimization objective is to minimize the makespan which means the time when all the jobs have been processed and is defined as follows:

$$Cmax = \max \{Tep_{jN_j^{opr}} \mid j \in J\} \tag{7}$$

where $Cmax$ is the maximum completion time of all jobs and $Tep_{jN_j^{opr}}$ can be calculated according to Eq. (8)-(13).

$$Tep_{j_2 i_2} = Tsp_{j_2 i_2} + t_{j_2 i_2 m_3}^P \tag{8}$$

where $X_{j_2 i_2 m_3} = 1$, $j_2 \in J, i_2 \in O_{j_2}^{opr}$ and $m_3 \in M$.

$$Tsp_{j_2 i_2} = \max\{Tet_{j_2 i_2}, Tep_{j_3 i_3}\} \tag{9}$$

where $Y_{j_3 i_3, j_2 i_2}^{m_3} = 1$, $j_2, j_3 \in J$, $i_2 \in O_{j_2}^{opr}$ and $i_3 \in O_{j_3}^{opr}$.

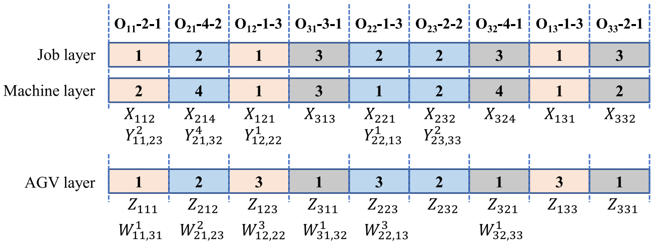
$$Tet_{j_2 i_2} = Tst_{j_2 i_2} + t_{d_2 d_3} \tag{10}$$

where d_2, d_3 are the position indexes of M_{m_2} and M_{m_3}, $X_{j_2 (i_2 - 1) m_2} = 1$, $X_{j_2 i_2 m_3} = 1$, $Z_{j_2 i_2 k} = 1$, $j_2 \in J$, $i_2 \in O_{j_2}^{opr}$ and $m_2, m_3 \in M$.

$$Tst_{j_2 i_2} = \max\{Tep_{j_2 (i_2 - 1)}, Tet'_{j_2 i_2}\} \tag{11}$$

where $j_2 \in J$ and $i_2 \in O_{j_2}^{opr}$.

$$Tet'_{j_2 i_2} = Tst'_{j_2 i_2} + t'_{d_1 d_2} \tag{12}$$



	O_{11}-2-1	O_{21}-4-2	O_{12}-1-3	O_{31}-3-1	O_{22}-1-3	O_{23}-2-2	O_{32}-4-1	O_{13}-1-3	O_{33}-2-1
Job layer	1	2	1	3	2	2	3	1	3
Machine layer	2	4	1	3	1	2	4	1	2
	X_{112} $Y^2_{11,23}$	X_{214} $Y^4_{21,32}$	X_{121} $Y^1_{12,22}$	X_{313}	X_{221} $Y^1_{22,13}$	X_{232} $Y^2_{23,33}$	X_{324}	X_{131}	X_{332}
AGV layer	1	2	3	1	3	2	1	3	1
	Z_{111} $W^1_{11,31}$	Z_{212} $W^2_{21,23}$	Z_{123} $W^3_{12,22}$	Z_{311} $W^1_{31,32}$	Z_{223} $W^3_{22,13}$	Z_{232}	Z_{321} $W^1_{32,33}$	Z_{133}	Z_{331}

Fig. 3. A chromosome example of FJCSP.

where d_1, d_2 are the position indexes of M_{m_1} and M_{m_2}, $W^k_{j_1i_1,j_2i_2} = 1$, $X_{j_2(i_2-1)m_2} = 1$, $X_{j_1i_1m_1} = 1$, $Z_{j_2i_2k} = 1$, $Z_{j_1i_1k} = 1$, $j_1, j_2 \in J$, $i_1 \in O^{opr}_{j_1}$, $i_2 \in O^{opr}_{j_2}$ and $m_1, m_2 \in M$.

$$Tst'_{j_2i_2} = Tet_{j_1i_1} \tag{13}$$

where $Z_{j_1i_1k} = 1$, $Z_{j_2i_2k} = 1$, $W^k_{j_1i_1,j_2i_2} = 1$, $j_1, j_2 \in J$, $i_1 \in O^{opr}_{j_1}$, $i_2 \in O^{opr}_{j_2}$ and $k \in A$.

Eq. (8) means that the ending time of the processing stage consists of the starting time and the processing time of the operation. Eq. (9) implies that the operation will begin to be processed after the job is transported to the machine and the previous processing task in the machine is completed. Eq. (10) means that the ending time of the transportation stage consists of the starting time and the transportation time on the AGV. Eq. (11) means that the job needs to be transported after the previous operation of the job is processed and the AGV reaches the machine processing the previous operation. Eq. (12) is the constraint of the ending time of the no-load stage. Eq. (13) implies that the AGV must complete the previous transportation task before taking the current operation.

D. Model of FJCSP

On the basis of the above analysis, the mathematical model can be summarized as follows:

$$\begin{aligned} &\min \ Cmax = \max\{Tep_{jN^{opr}_j} \mid j \in J\} \\ &s.t. \text{ Eqs. } (1) - (6) \end{aligned} \tag{14}$$

Remark 2: The scale of an FJCSP can be described as a tuple $\langle N_J, N_O, N_M, N_A\rangle$ consisting of the numbers of jobs, machines, operations and AGVs.

E. Encoding Method

In this section, a three-layer representation of the solution is adopted. The genes in the first layer (operation layer) are responsible for the expression of the processing and transportation sequence of all operations. The genes in the second layer (machine layer) are the indexes of machines to process the operations at corresponding positions in the operation layer. The genes in the third layer (AGV layer) are the indexes of AGVs that transport the corresponding jobs. The encoding example of an instance with the scale of $\langle 3, 9, 4, 3\rangle$ is shown in Fig.3. The complete scheduling scheme of this example can be described as a sequence of combinations about the operations and the assignment of machines and AGVs (O_{ji}, m, k). Each (O_{ji}, m, k) means that O_{ji} is assigned to the M_m and is transported by the A_k. Therefore, the scheduling scheme can be described as follows: $(O_{11}, 2, 1)$, $(O_{21}, 4, 2)$, $(O_{12}, 1, 3)$, $(O_{31}, 3, 1)$, $(O_{22}, 1, 3)$, $(O_{23}, 2, 2)$, $(O_{32}, 4, 1)$, $(O_{13}, 1, 3)$, $(O_{33}, 2, 1)$.

F. Decoding Method

According to the three-layer chromosome, the decision variables can be determined. X_{jim} and Z_{jik} can be determined through the combination (O_{ji}, m, k) in the chromosome. In addition, $Y^m_{j_1i_1,j_2i_2}$ and $W^k_{j_1i_1,j_2i_2}$ can be assigned by the sequential relationship between combinations. For the solution in Fig. 3, the decision variables which will be set to 1 are shown below the chromosome. According to the combination $(O_{21}, 4, 2)$ of the 2nd operation in the chromosome, $X_{214} = 1$ and $Z_{212} = 1$. Because the operation to be processed next by M_4 is O_{32}, $Y^4_{21,32} = 1$. Because the operation to be transported next by A_2 is O_{23}, $W^2_{21,23} = 1$. The more direct decoding procedures are shown in Algorithm 1.

Algorithm 1 Decoding Process

```
Input: The chromosome of a solution $x$;
Output: The makespan of $x$;
$Tej_j \Leftarrow 0, \forall j \in J$; % Set the initial time on jobs;
$Dej_j \Leftarrow 0, \forall j \in J$; % Set the position index of each job;
$Tem_m \Leftarrow 0, \forall m \in M$; % Set the initial time on machines;
$Tea_k \Leftarrow 0, \forall k \in A$; % Set the initial time on AGVs;
$Dea_k \Leftarrow 0, \forall k \in A$; % Set the position index of each AGV;
for $g = 1$ to $N_O$ do:
    Extract the combination $(O_{ji}, m, k)$ of the $g$-th column in the chromosome;
    $Tst'_{ji} = Tea_k$ according to Eq.(13);
    $Tet'_{ji} = Tst'_{ji} + t'_{Dea_kDej_j}$ according to Eq.(12);
    $Tst_{ji} = \max\{Tej_j, Tet'_{ji}\}$ according to Eq.(11);
    $Tet_{ji} = Tet'_{ji} + t_{Dej_jm}$ according to Eq.(10);
    $Tsp_{ji} = \max\{Tet_{ji}, Tem_m\}$ according to Eq.(9);
    $Tep_{ji} = Tsp_{ji} + t^P_{jim}$ according to Eq.(8);
    Update $Tej_j \Leftarrow Tep_{ji}$, $Dej_j \Leftarrow m$, $Tem_m \Leftarrow Tep_{ji}$, $Tea_k \Leftarrow Tet_{ji}$, $Dea_k \Leftarrow m$;
end for
Calculate the makespan according to Eq. (7).
```

III. Hybrid Algorithm for FJCSP

In this paper, we propose a multi-view modeling-based hybrid algorithm, in which EDA with a **probability model** and ACO with a **pheromone model** are designed to conduct the global exploration cooperatively. In addition, a local search based on the critical path method is used for local exploitation.

A. Optimization Methods Based on Statistical Model

The model mentioned in the proposed algorithm refers to statistical model that attempts to estimate some kind of characteristics of the optimal solution. The statistical model

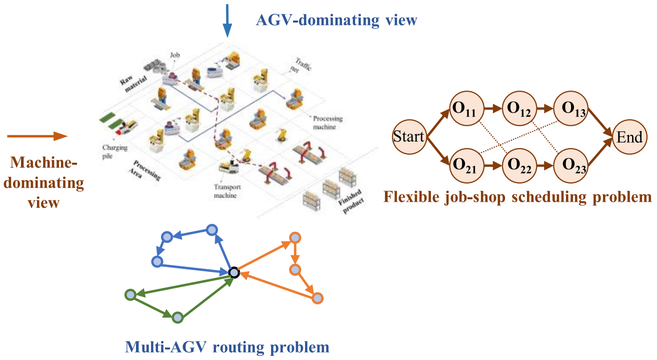


Fig. 4. The two views of FJCSP.

can help to interpret the results of previous optimization steps and to plan the next ones [27]. The feature selection of the solution is closely related to the establishment of the statistical model. Therefore, the feature selection is important and only the models reflecting key features of the optimal solution are necessary.

For complex problems, a single model may be weak because it does not reflect completely or accurately the key features. Compared with a single model, multiple models with different key features can bring the following benefits:

1) Multiple models can characterize the solution more comprehensively and reduce dependence on a single model;
2) Multiple models can improve population diversity and reduce the possibility of premature convergence.

The modeling method from multiple views (perspectives) can be applied to FJCSP as shown in Fig. 4. On the one hand (machine-dominating view), FJCSP can be regarded as an extension of FJSP and most scholars designed their algorithms from this view. On the other hand (AGV-dominating view), the routing problem of AGVs may be the main issue in some cases when the transportation stages take more time than the processing stages. It can be found that the algorithms designed from any single view may be weak for these instances due to the difference of key features. Therefore, we introduced EDA and ACO for the two features to solve FJCSP. In this paper, we design a novel probability model in EDA and a joint pheromone model in ACO to grasp the key features of elite solutions from the two views, respectively. In the proposed hybrid algorithm, EDA and ACO generate solutions independently and achieve cooperation by sharing elites. Compared with previously proposed algorithms such as GAs, the modeling-based method establishes a statistical model for the distribution of the entire population, describing the direction of evolution through this model. It represents a macroscopic level simulation. Furthermore, the modeling-based method enables more convenient utilization of problem knowledge.

B. Probability Model in EDA

The probability model can be described as three matrixes, i.e., operation-related probability matrix (OPM, P^{opr}), machine-related probability matrix (MPM, P^{mac}) and AGV-related probability matrix (APM, P^{agv}) to generate the three-layer representation of a solution in Section II-E. The operation layer can be generated from the first position to the last by sampling the OPM directly. However, the machine layer and the AGV layer need to be generated according to the heuristic rule selected by the MPM and the APM. These heuristic rules for machines and AGVs which can obtain good-quality solutions are introduced, and the heuristic rules including 7 machine selection rules (MSR) and 7 AGV selection rules (ASR) can be referred to the recent research [28] as shown in Table II. ASR1, ASR2, MSR1 and MSR2 are the selection rules for $O_{j1}(j \in J)$.

TABLE II
SELECTION RULES OF AGVS AND MACHINES FOR PROCESSING $O_{j(i+1)}$

Rule	Description
ASR1	Select the AGV that reaches the material area the earliest.
ASR2	Select the AGV that is available the earliest.
ASR3	Select the AGV that reaches the current machine the latest before O_{ji} is finished, if one exists. Otherwise, select the AGV that reaches the machine the earliest.
ASR4	Select the AGV that reaches the current machine the earliest after O_{ji} is finished, if one exists. Otherwise, select the AGV that reaches the machine the latest.
ASR5	Select the AGV that minimizes waiting time for each other.
ASR6	Select the AGV that reaches the machine the earliest.
ASR7	Select the AGV that is available the earliest.
MSR1	Select the machine that finishes the operation the earliest.
MSR2	Select the machine that is available the earliest.
MSR3	Select the machine that is available the latest before J_j reaches it, if one exists. Otherwise, select the machine that is available the earliest.
MSR4	Select the machine that is available the earliest after J_j reaches it, if one exists. Otherwise, select the machine that is available the latest.
MSR5	Select the machine that minimizes waiting time for each other.
MSR6	Select the machine that finishes processing O_{ij} the earliest.
MSR7	Select the machine that is available the earliest.

The probability matrixes can be described as follows:

$$P^{opr}(l) = [p^{opr}_{ij}(l)]_{N_O \times N_J} \tag{15}$$

$$P^{mac}(l) = [p^{mac}_{j}(l)]_{N_{RM} \times 1} \tag{16}$$

$$P^{agv}(l) = [p^{agv}_{j}(l)]_{N_{RA} \times 1} \tag{17}$$

where N_{RM} is the number of the machine scheduling rules and N_{RA} is the number of the AGV scheduling rules; the element $p^{opr}_{ij}(l)$ of P^{opr} means the probability that J_j appears at position i in the first layer at generation l. The element $p^{mac}_{j}(l)$ of P^{mac} represents the probability that the genes in the second layer for processing operations are selected by the j-th MSR at generation l. The element $p^{agv}_{j}(l)$ of P^{agv} represents the probability that the genes in the third layer for processing operations are selected by the j-th ASR at generation l.

Without prior knowledge, these probability matrices are uniformly initialized, and the update mechanisms of the three probability matrixes are as follows:

$$p^{opr}_{ij}(l+1) = (1-\lambda^{eda})p^{opr}_{ij}(l) + \frac{\lambda^{eda}}{N_s}\sum_{s=1}^{N_s} I^{opr}_{sij} \tag{18}$$

$$p^{mac}_{j}(l+1) = (1-\lambda^{eda})p^{mac}_{j}(l) + \frac{\lambda^{eda}}{N_s}\sum_{s=1}^{N_s} I^{mac}_{sj} \tag{19}$$

$$p^{agv}_{j}(l+1) = (1-\lambda^{eda})p^{agv}_{j}(l) + \frac{\lambda^{eda}}{N_s}\sum_{s=1}^{N_s} I^{agv}_{sj} \tag{20}$$

where N_s represents the number of superior individuals and λ^{eda} is the learning rate. I_{sij}^{opr}, I_{sj}^{mac} and I_{sj}^{agv} are the indicators as follows:

$$I_{sij}^{opr} = \begin{cases} 1, & \text{if the } j\text{-th job appears at position } i \\ & \text{in the } s\text{-th solution,} \\ 0, & \text{else.} \end{cases} \tag{21}$$

$$I_{sj}^{mac} = \begin{cases} 1, & \text{if the } j\text{-th MSR is applied in the} \\ & s\text{-th solution,} \\ 0, & \text{else.} \end{cases} \tag{22}$$

$$I_{sj}^{agv} = \begin{cases} 1, & \text{if the } j\text{-th ASR is applied in the} \\ & s\text{-th solution,} \\ 0, & \text{else.} \end{cases} \tag{23}$$

The pseudo-code of sampling a solution by EDA is shown in Algorithm 2.

Algorithm 2 Sampling a Solution by EDA

```
Input: $P^{opr}$, $P^{agv}$, $P^{mac}$;
Output: A solution $x$;
Set $g \Leftarrow 1$;
for $l = 1$ to $N_J$ do:
    $\varepsilon_l = 0$ % Initialize the operation index of each job;
end for
while $g \le N_O$ do: % Obtain the operation sequence;
    for $l = 1$ to $N_J$ do:
        $R(l) = \begin{cases} p_{gl}^{opr}, & \text{if } \varepsilon_l < N_l^{opr}, \\ 0, \text{else.} \end{cases}$ % Build roulette;
    end for
    $Normalize(R)$ % Normalize the roulette;
    Obtain a job index $j$ on the $g$-th position in operation layer by wheeling $R$;
    Set $\varepsilon_j \Leftarrow \varepsilon_j + 1$, $g \Leftarrow g + 1$;
end while
Obtain an ASR by sampling $P^{agv}$ in roulette wheel;
Obtain an MSR by sampling $P^{mac}$ in roulette wheel;
Set $G \Leftarrow \{O_{ji} \mid j \in J, i \in N_j^{opr}\}$, $g \Leftarrow 1$;
while $G \neq \emptyset$ do:
    Retrieve the $g$-th operation $O_{ji}$ to be assigned next in the operation layer;
    Assign $A_k$ for $O_{ji}$ using ASR shown in Table II;
    Assign $M_m$ for $O_{ji}$ using MSR shown in Table II;
    Update $Tst'_{ji}$, $Tet'_{ji}$, $Tst_{ji}$, $Tet_{ji}$, $Tsp_{ji}$ and $Tep_{ji}$ of $O_{ji}$ for $J_j$, $M_m$ and $A_k$ as shown in Algorithm 1;
    Set $G \Leftarrow G/\{O_{ji}\}$, $g \Leftarrow g + 1$.
end while
```

C. Pheromone Model in ACO

ACO was originally proposed by Dorigo, Maniezzo and Colorni in the 1990s. In the process of studying ant foraging, they found that ant colony as a whole can reflect swarm intelligence for optimizing by simple behaviors of each ant, including laying pheromone to the environment and deciding the next step based on the pheromone.

1) Pheromone Design: As the key element of ACO for solving the problem in this paper, the pheromone needs to be redesigned to fit the problem feature. Considering the operation flows of jobs and the paths of AGVs in FMS, two kinds of pheromones are designed: job-related pheromone $\tau_{jid_1m}^{job}(l)$ and AGV-related pheromone $\tau_{d_1d_2}^{agv}(l)$. The element in job-related pheromone $\tau_{jid_1m}^{job}(l)$ represents the pheromone amount deposited for J_j from the current position D_{d_1} to the target position D_m to process O_{ji} at generation l. The element in AGV-related pheromone $\tau_{d_1d_2}^{agv}(l)$ represents the total pheromone deposited for AGVs from the current position D_{d_1} to D_{d_2} at generation l. They can be initialized as follows:

$$\tau_{jid_1m}^{job}(0) = \frac{1}{\mid D \mid}, \quad j \in J,\ i \in N_j^{opr},\ d_1 \in D, m \in M \tag{24}$$

$$\tau_{d_1d_2}^{agv}(0) = \frac{1}{\mid D \mid}, \quad d_1, d_2 \in D \tag{25}$$

where the initial pheromone value on each path is set as $1/\mid D \mid$ uniformly.

2) Transition Strategy: According to the two pheromone models, a set of combinations of job-machine-AGV are selected step by step until all the operations are assigned. In this way, each ant will select one machine and one AGV for one operation in each step.

The selection of the combination (O_{ji}, m, k) depends on the joint factor of the two pheromones. Define the joint pheromone factor as follows:

$$\tau_{jimk}^{joint}(l) = (\tau_{d_kd_1}^{agv}(l) + \tau_{d_1m}^{agv}(l)) \times \tau_{jid_1m}^{job}(l) \tag{26}$$

where d_k is the current position index of A_k when A_k starts to transport O_{ji}, d_1 is the current position index of J_j and m is the position index of the target machine for processing O_{ji}.

The probability for selecting the combination (O_{ji}, m, k) by roulette wheel is

$$p_{jimk} = \frac{\tau_{jimk}^{joint}}{\sum\limits_{O_{ji} \in G} \sum\limits_{m \in \Omega_{ji}} \sum\limits_{k \in A} \tau_{jimk}^{joint}}, \tag{27}$$

where p_{jimk} is the probability of selecting A_k to transport O_{ji} to M_m, and G is the set of operations to be processed next. The pseudo-code of sampling a solution by ACO is shown in Algorithm 3.

Remark 3: All the optional combinations about all the operations to be processed, the available machines and AGVs are considered in Eq. (27). For example, if O_{12} and O_{32} need to be processed and the number of the available AGVs is N_A, then the total number of all the optional combinations are $\mid \Omega_{12} \mid \times N_A + \mid \Omega_{32} \mid \times N_A$.

3) Pheromone Trail Update Rule: The pheromone trail update rule of the two matrixes for route selection is shown as follows:

$$\tau_{jid_1m}^{job}(l+1) = \xi^{aco}\tau_{jid_1m}^{job}(l) + (1 - \xi^{aco})\Delta\tau_{jid_1m}^{job}(l) \tag{28}$$

where $\Delta\tau_{jid_1m}^{job}(l) = \sum_{s=1}^{N_s} I_{sjid_1m}^{job}/N_s$ is the pheromone amount added by N_s superior individuals for J_j from D_{d_1} to D_m for processing O_{ji}, and ξ^{aco} is the volatilization rate of ACO.

$$\tau_{d_1d_2}^{agv}(l+1) = \xi^{aco}\tau_{d_1d_2}^{agv}(l) + (1 - \xi^{aco})\Delta\tau_{d_1d_2}^{agv}(l) \tag{29}$$

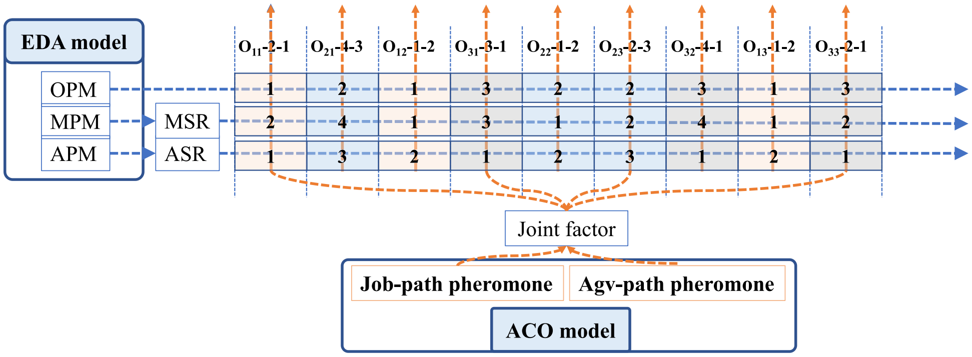


Fig. 5. An Illustration for generating a solution by EDA/ACO.

where $\Delta\tau_{d_1d_2}^{agv}(l) = \sum_{s=1}^{N_s} I_{sd_1d_2}^{agv}/N_s$ is the pheromone amount added by N_s superior individuals for AGVs from D_{d_1} to D_{d_2}.

Algorithm 3 Sampling a Solution by ACO

```
**Input**: $\tau_{jid_1m}^{job}$, $\tau_{d_1d_2}^{agv}$;
**Output**: A solution $x$;
Set $G \Leftarrow \{O_{j1} \mid j \in J\}$;
**while** $G \neq \emptyset$ **do**:
    **for** $O_{ji}$ in $G$ **do**:
        **for** $k$ in $A$ **do**:
            **for** $m$ in $\Omega_{ji}$ **do**:
                Calculate the joint factor $\tau_{jimk}^{joint}$ about transporting $O_{ji}$ to $M_m$ by $A_k$ according to (26);
    Select the combination of $(O_{ji}, m, k)$ by roulette wheel according to the probability matrix in (27);
    $G \Leftarrow G/\{O_{ji}\}$;
    If $i \neq N_j^{opr}$, then $G \Leftarrow G \cup \{O_{j(i+1)}\}$.
**end while**
```

In proposed hybrid algorithm, both EDA and ACO can sample a complete solution independently. However, they generate solutions from different aspects. The detailed illustration in Fig. 5 shows the sampling procedures of EDA and ACO. For EDA, the operation sequence in the first layer of the chromosome, MSR and ASR are obtained by sampling by OPM, MPM and APM, respectively. And then the schemes for machines and AGVs are determined according to MSR, ASR and the operation sequence step by step. For ACO, the ant will select an operation to be processed next, an optional machine and an AGV simultaneously in each step according to the pheromones.

D. EDA/ACO Selection Ratio

The use proportion of EDA and ACO needs to be adjusted dynamically according to their performance. The reason can be summarized into two points:

1) Their performance may be different for various cases;
2) Their performance may be varying with search stages.

Therefore, we define $\lambda(l) \in [0.1, 0.9]$ as the selection ratio of EDA and ACO in the l-th iteration. Initially, set $\lambda(0) = 0.5$ without prior knowledge, which means an equal chance of selecting EDA and ACO. The update rule of $\lambda(l)$ is shown as follows:

$$\lambda(l+1) = (1-\omega)\cdot\lambda(l) + \omega\cdot\frac{N_E}{N_B} \tag{30}$$

where ω denotes the learning rate, N_E is the number of individuals generated by EDA in elite individuals and N_B is the number of elite individuals.

E. Selection Mechanism

In order to maintain the population diversity and keep the best individuals in the population, a multi-strategy selection operation needs to be designed. Firstly, the elite reservation is embedded into the selection mechanism. The individual with the best performance will always be selected for the next generation. Secondly, a survival ratio parameter P_s means the proportion of the individuals selected from the population into the next generation in the whole population. According to the parameter, the binary tournament selection strategy is adopted to select a specified number of individuals for the next generation. In order to supplement the population diversity, a reproduction strategy is adopted to maintain the previous population size and improve the population diversity.

F. Local Search Based on Critical Path Method

Although EDA and ACO can pay more attention to the characteristics of problems for global exploration, the local exploitation in solution space still needs to be implemented by a local search [29]. In the traditional research for job shop scheduling problems, many local search methods based on the critical path method (CPM) have been studied widely and have good performance [30]. There are no idle time and no possible overlap time between any two adjacent processes in the critical path. *No idle time* means no time can be compressed and *no possible overlap time* means there is a strict logical relationship about the task sequence. The length of the critical path is equal to the makespan.

For the simultaneous scheduling problem, a critical path may include the processing and transportation blocks, called *machine-block*s and *AGV-block*s, respectively [31]. For example in Fig. 6, three machine-blocks and three AGV-blocks make up of the critical path P^C, $S_{11}^T \Rightarrow S_{11}^P \Rightarrow S_{12}^T S_{22}^{T'} S_{22}^T \Rightarrow S_{22}^P \Rightarrow S_{23}^T \Rightarrow S_{23}^P S_{33}^P$. From the perspective of the operation,

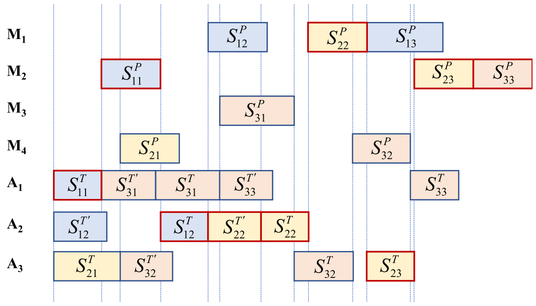


Fig. 6. An example of a critical path in gantt chart.

S_{11}^{T} and S_{11}^{P} of O_{11}, S_{12}^{T} of O_{12}, $S_{22}^{T'}S_{22}^{T}$ and S_{22}^{P} of O_{22}, S_{23}^{T} and S_{23}^{P} of O_{23}, S_{33}^{P} of O_{33} make up the critical path P^C. Thus, for O_{ji} in the critical path, there may be the following three kinds of cases:

Case1: $S_{ji}^{T} \in P^C$ & $S_{ji}^{P} \in P^C$ means both the transportation stage and processing stage of O_{ji} are in the critical path;

Case2: $S_{ji}^{T} \notin P^C$ & $S_{ji}^{P} \in P^C$ means only the processing stage of O_{ji} is in the critical path;

Case3: $S_{ji}^{T} \in P^C$ & $S_{ji}^{P} \notin P^C$ means only the transportation stage of O_{ji} is in the critical path.

In order to reduce the length of the critical path (to minimize makespan), three operators for the above three cases of an O_{ji} selected randomly on critical path are proposed to define the neighborhood structures for local search:

$Opr1$ for case1: Exchange three-layer genes with a random adjacent operation of O_{ji};

$Opr2$ for case2: Replace randomly the gene in the machine layer of O_{ji} with another optional machine;

$Opr3$ for case3: Replace randomly the gene in the AGV layer of O_{ji} with another optional AGV.

Algorithm 4 Local Search Based on CPM

```
Input: a solution $x$ and the number of iteration $Iter_{max}^{LS}$;
Output: An improved solution $x'$;
Set $g = 0$;
while $g < Iter_{max}^{LS}$ do
    Find the critical path $P_C$ of $x$;
    Select an operation $O_{ji}$ in $P_C$ randomly;
    if $S_{ji}^{T} \in P^C$ and $S_{ji}^{P} \in P^C$ then
        Execute $Opr1$ for $x$ to generate $x'$;
    if $S_{ji}^{P} \in P^C$ and $S_{ji}^{T} \notin P^C$ then
        Execute $Opr2$ for $x$ to generate $x'$;
    if $S_{ji}^{T} \in P^C$ and $S_{ji}^{P} \notin P^C$ then
        Execute $Opr3$ for $x$ to generate $x'$;
    Calculate the makespan value $C'_{max}$ for $x'$;
    if $C'_{max} < C_{max}$ then
        $x \Leftarrow x'$;
        $C_{max} \Leftarrow C'_{max}$;
    $g \Leftarrow g + 1$;
end while
$x' \Leftarrow x$.
```

TABLE III
TIME COMPLEXITY ANALYSIS

Procedure	Time complexity
Generating a new population	$O(N \cdot N_J^2)$
Selecting elite individuals	$O(N_s)$
Updating models of EDA and ACO	$O(N_s \cdot N_J^2)$
Improving a solution by local search	$O(Iter_{max}^{LS})$
Updating ratio of selecting EDA/ACO	$O(N_s)$

The pseudo-code of CPM-based local search is shown in Algorithm 4. In every generation, the local search will be applied $Iter_{max}^{LS} = 200$ times on the best individual of the current population, and the better solution found by local search will replace the old one. Although only a few genes in the new solution are changed by local search, the solution still needs to be re-evaluated completely, which costs the same time as evaluating a solution generated by EDA or ACO.

G. Procedure of Proposed Algorithm

In the first stage, the search space is sampled according to the selection ratio λ of EDA and ACO to generate a new population by Algorithm 2 and Algorithm 3. Then, the elite group is selected for the following updating process of probability matrixes and pheromone matrixes. Then, the CPM-based local search is carried out to improve the best individuals by Algorithm 4. The probability model of EDA and the pheromone model of ACO are updated by the information of the elite group. In addition, λ is also updated by the performance of EDA and ACO according to (30). Finally, whether the evolutionary loop is stopped is determined by the termination criterion.

H. Computational Complexity Analysis

At each generation of the proposed algorithm, the time complexity analysis is presented in Table III.

The time complexity of generating a new population by ACO or EDA is $O(N \cdot N_J^2)$. For the binary tournament selection of the proposed algorithm, there is a computational complexity of $O(N_s)$. Then, about the update process for the probability model of EDA and pheromone in ACO, it has a computational complexity of $O(N_s \cdot N_J^2)$. For the local search based on CPM, the computational complexity is $O(Iter_{max}^{LS})$. Therefore, the main time complexity is introduced by generating a new population and updating the models of EDA and ACO.

IV. NUMERICAL RESULTS AND COMPARISONS

A. Design of Experiments

In this section, we design three experiments for analyzing the proposed algorithm. In the first experiment, the Taguchi method is carried out for setting key parameters of the proposed algorithm (namely EDA_ACO_LS) for the different instances. In the second experiment, the following algorithms are run 30 times for solving different instances to verify the effectiveness and advantages of the proposed algorithm:

1) sEDA: The single EDA which uses the same probability model and parameter values in EDA_ACO_LS;

2) sACO: The single ACO which uses the same pheromone model and parameter values in EDA_ACO_LS;
3) EDA_ACO: The hybrid algorithm combining probability model and pheromone model without the local search in EDA_ACO_LS;
4) RGA [32]: A multi-objective hybrid genetic algorithm combining a selection strategy for AGVs, in which probability of crossover $CR = 0.8$ and probability of mutation $MR = 0.4$;
5) PDE [33]: A differential evolution with a machine selection heuristic and a vehicle assignment heuristic in which $CR = 0.5$ and scaling factor $F = 0.8$;
6) GATS [34]: A hierarchical procedure that combines a genetic algorithm and a tabu search procedure, whose $CR = 1$ and $MR = 1$;
7) BRKGA [28]: An operation-based multi-start biased random key genetic algorithm coupled with greedy heuristics to select the machine processing each operation and the vehicles transporting the jobs to operations, whose selection proportion of elites $P_S = 0.2$, $MR = 0.1$ and elite inheritance probability $P_E = 0.7$.

The above comparison algorithms proposed in previous literature are calibrated for the benchmark reported in Bilge and Ulusoy [7] and BRKGA can also be calibrated according to the percentage gaps. All the calibration results are shown in the attachment. Finally, a case study in a real-world intelligent hardware factory is carried out for the actual production.

Three benchmarks are carried out for discussion and analysis, including small-sized instances by Deroussi and Norre [35], small-sized instances by Kumar et.al [33] and small-medium-large-sized instances by Homayouni et.al [28]. All of these instances are collected from https://fastmanufacturingproject.wordpress.com/2019/04/11/fjspt-instances/. These instances are named by a set of strings. For example, ***Deroussi_6_15_8_2*** means the instance proposed by *Deroussi* has ***6*** jobs, ***15*** operations, ***8*** machines and ***2*** AGVs.

In this paper, each instance for the following experiments will be run 30 times independently. All of the following experiments for different algorithms are carried out with AMD Ryzen 7 4800H (2.90GHz) with Radeon Graphics (RAM_32G). All algorithms will be terminated after evaluating solutions the maximum number NSE times shown in Eq. (31) to keep fairness. In addition, the number of evaluations during the local search in EDA_ACO_LS is also considered into NSE.

$$NSE = 100 \times N_O \times N_M \times N_A \tag{31}$$

B. Parameter Setting

The proposed algorithm has the following key parameters: 1) population size (N); 2) selection ratio of the elite group (P_s); 3) the learning rate of probability model in EDA (λ^{eda}); 4) the evaporation factor of pheromone in ACO (ξ^{aco}); 5) the learning rate of the selection ratio (ω). In this paper, the Taguchi method of design-of-experiment is applied to discuss the influence of the five parameters [36]. Among all the parameters of the problem, t/p is important to represent the relative size between transportation time and processing time in an instance [7]. Four levels of each parameter are set, i.e., $N \in \{100, 200, 300, 400\}$, $P_s \in \{0.1, 0.2, 0.3, 0.4\}$, $\lambda^{eda} \in \{0.2, 0.4, 0.6, 0.8\}$, $\xi^{aco} \in \{0.2, 0.4, 0.6, 0.8\}$, $\omega \in \{0.2, 0.4, 0.6, 0.8\}$. Two instances, ***Homayouni_11_44_8_2***(Exp1) and ***Homayouni_10_55_6_2***(Exp2), are selected to test these parameters, whose t/p are equal to 0.04 and 2.25, respectively.

TABLE IV
ORTHOGONAL ARRAY AND RV VALUES

Experiment	Factor Level					Makespan	
Number	N	P_s	λ^{eda}	ξ^{aco}	ω	Exp1	Exp2
1	100	0.1	0.2	0.2	0.2	1156.3	192.5
2	100	0.2	0.4	0.4	0.4	1158.2	192.1
3	100	0.3	0.6	0.6	0.6	1156.7	192.8
4	100	0.4	0.8	0.8	0.8	1166.0	195.3
5	200	0.1	0.4	0.6	0.8	1143.4	191.8
6	200	0.2	0.2	0.8	0.6	1142.4	194.6
7	200	0.3	0.8	0.2	0.4	1131.5	190.5
8	200	0.4	0.6	0.4	0.2	1144.7	192.6
9	300	0.1	0.6	0.8	0.4	1141.2	190.6
10	300	0.2	0.8	0.6	0.2	1132.3	190.3
11	300	0.3	0.2	0.4	0.8	1130.5	192.7
12	300	0.4	0.4	0.2	0.6	1137.1	193.0
13	400	0.1	0.8	0.4	0.6	1122.2	190.0
14	400	0.2	0.6	0.2	0.8	1133.7	191.3
15	400	0.3	0.2	0.8	0.2	1135.0	191.9
16	400	0.4	0.2	0.6	0.4	1135.0	192.5

TABLE V
RESPONSE TABLE FOR EXP1

Factor Level	N	P_s	λ^{eda}	ξ^{aco}	ω
1	1159	1141	1141	1140	1142
2	1140	1142	1143	**1139**	1141
3	1135	**1138**	1144	1142	**1140**
4	**1131**	1146	**1138**	1146	1143
Delta	29	8	6	7	4
Rank	1	2	4	3	5

According to the Taguchi method, a set of tests with different parameter values are carried out to establish the orthogonal array $L_{16}(4^5)$. For each test, the proposed algorithm is run 30 times independently and the average fitness value (makespan) is calculated and filled into Table IV. Then, the response average values of each factor level for different instances are shown in Table V and Table VI. It can be found that the major differences between instances with different t/p are the P_s and ω. From the Rank vectors in Table V and Table VI, it is obvious that the population size (N) and the selection ratio of the elite group (P_s) are more important than the other parameters. Considering the different t/p levels of different instances, the two combinations of the five parameters for different instances are introduced and shown in Table VII according to the factor levels in Table V and Table VI. In general, the difference in parameter selection of the algorithm under different calculation examples is acceptable.

C. Experimental Analysis

First of all, a comparisons criterion under significant differences needs to be stated: if $Alg1$ performs better than $Alg2$, one of the following conditions needs to be met: 1) $Mean(Alg1) < Mean(Alg2)$; 2) $Mean(Alg1) = Mean(Alg2)$ and $Std(Alg1) < Std(Alg2)$,

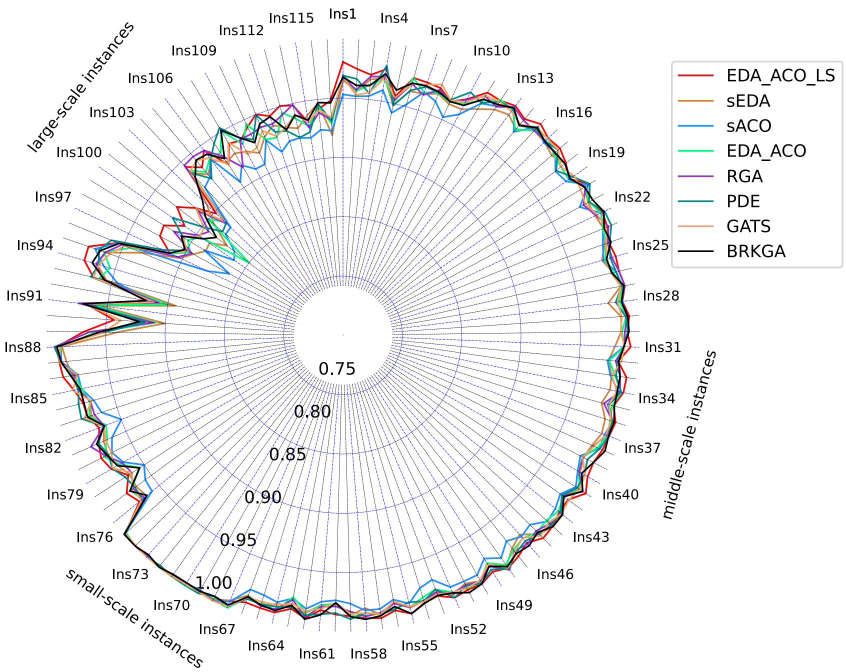


Fig. 7. The radio map of the gaps with best results ($\rho = 1.0$) of different algorithms for all instances.

TABLE VI
RESPONSE TABLE FOR EXP2

Factor Level	N	P_s	λ^{eda}	ξ^{aco}	ω
1	193.2	**191.3**	193.1	**191.8**	191.8
2	192.4	192.1	192.2	**191.8**	**191.4**
3	191.7	192.0	191.8	191.9	192.6
4	**191.4**	193.3	**191.5**	193.1	192.8
Delta	1.7	2.1	1.5	1.3	1.3
Rank	2	1	3	5	4

TABLE VII
SUGGESTED COMBINATION OF PARAMETERS

Instance	N	P_s	λ^{eda}	ξ^{aco}	ω
$t/p \leq 0.5$	400	0.3	0.8	0.4	0.6
$t/p > 0.5$	400	0.1	0.8	0.4	0.4

where $Alg1$ and $Alg2$ means two different algorithms, $Min(Alg1)$, $Mean(Alg1)$ and $Std(Alg1)$ indicate the minimum value, the mean value and the standard deviation of the vector of the best makespan found by $Alg1$ in the runs of 30 times. Then, the rank of an algorithm for an instance is defined as the number of other comparison algorithms which the algorithm performs worse than. For example, the rank of EDA_ACO_LS among all comparison algorithms for an instance is Rank5 means that 5 algorithms outperform EDA_ACO_LS. Obviously, the higher rank of an algorithm, the better its performance.

Due to space constraints, the experiment results for various instances with different scales and t/p are briefly shown in a radio map as Fig. 7 and more details are listed in the attachment. In the radio map of Fig. 7, a data point can be represented as (θ, ρ). The *angular coordinate* θ indicates the instance index and the *radial coordinate* ρ is a measure on the tending to the best makespan Mkb for an algorithm Alg, which is calculated by $\rho = 1 - Gap$, where $Gap = [Mean(Alg) - Mkb]/Mkb \times 100\%$. The more detail analysis will be shown in the following sections. A non-parameter statistical significance test method, i.e., the *Mann-Whitney U test* (Rank Sum Test, Rst) [37] at a significance level of 0.05 was used to analyze

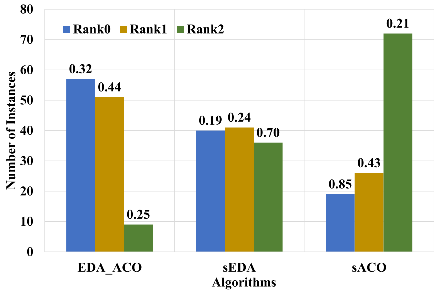


Fig. 8. The distribution of performance levels among sEDA, sACO and EDA_ACO (The average t/p of the instances in different ranks are shown above histograms).

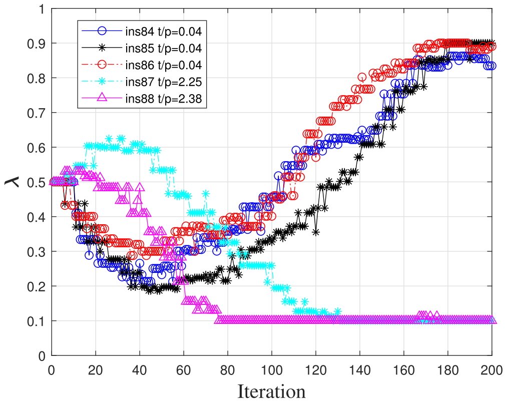


Fig. 9. Change of the selection ratio λ.

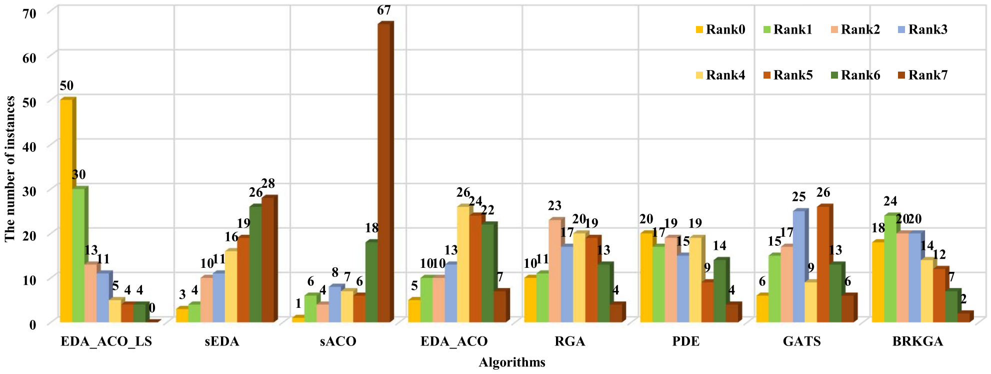


Fig. 10. The distribution of performance ranks of all comparison algorithms for all instances.

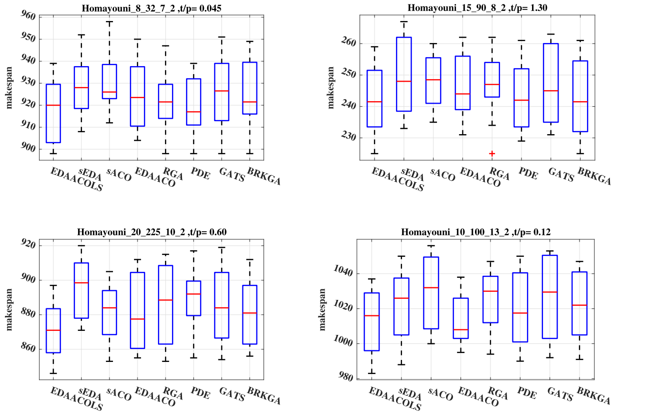


Fig. 11. Makespan distribution of different algorithms in classical instances.

the statistical significance of the results. The test results are also shown in attachment. In addition, we find new bounds of three instances including *Ins85*, *Ins86* and *Ins88*, and the solutions are listed in Table IX of Appendix.

1) Effect of the Combination of EDA and ACO: In order to verify the effectiveness of the combination of the two models, EDA_ACO is compared with sEDA and sACO for solving all the instances. The statistical information about the performance ranks among the three algorithms and the number of instances on the corresponding ranks of each algorithm is shown in Fig. 8. In addition, the average values of t/p in different ranks are also shown above the corresponding blocks to analyze the influence of t/p on these algorithms. For example, the blue block of EDA_ACO indicates that the number of instances on which EDA_ACO performs the best is 57 and the average t/p of these instances is equal to 0.32. From the figure, the following points can be summarized:

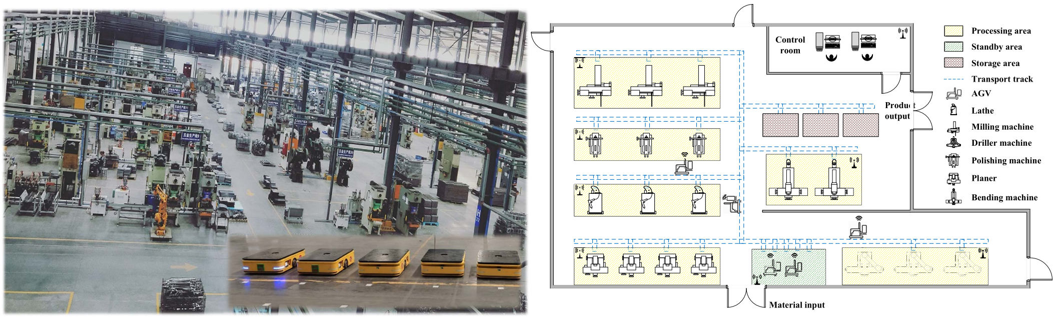
1) EDA_ACO performs the best in most of the instances, which means that the method combining EDA and ACO is more effective than any single-model method;
2) The average t/p values of sEDA in Rank1 and Rank2 are significantly less than sACO, which means sEDA performs better than sACO if t/p is small, and otherwise, sACO performs better than sEDA.

These conclusions become another basis for setting the algorithm parameters according to the problem parameter t/p. Despite the above points, we still cannot clearly define which model performs better than the other one in a certain instance. In the analysis for the selection ratio, it can be found the combination of EDA and ACO is still necessary. In addition, EDA_ACO, sEDA and sACO still need to be improved due to the lack of a local exploration mechanism.

2) Effect of Local Search: To verify the effectiveness of the local search proposed, EDA_ACO and EDA_ACO_LS are compared for solving all the instances. From Fig.7, it can be found that EDA_ACO_LS performs better than EDA_ACO in more than 50 instances at least. The obvious advantage indicates that the proposed local search can greatly improve the solution under global search by EDA and ACO.

3) Effect of Selection Ratio: Fig. 9 shows the change curves of the selection ratio λ with iterations for different instances including $Ins84$, $Ins85$, $Ins86$, $Ins87$ and $Ins88$. For $Ins88$, it can be found that EDA was mainly used to generate individuals in the early iterations, but the ACO performed better than EDA in the late iteration. The diversified trend of these curves indicates that this parameter can be adjusted adaptively according to different instances and search stages to ensure the performance of the proposed algorithm. This adaptive adjustment, coupled with innovative and effective modeling-based methods, has brought about a significant improvement in solution efficiency. In summary, the hybrid mechanisms in EDA_ACO_LS, including hybrid modeling and local search operator, all contribute to the advantages of the hybrid algorithm.

4) Comparison With State-of-the-Art Algorithms: The statistics of comparison experiments are visually displayed in Figs. 7 and 10. From Fig. 7, EDA_ACO_LS is closer to the edge of radar image, which means the average performance in 30 runs of EDA_ACO_LS is better than the other algorithms. From Fig. 10, it obvious that the proposed algorithm performs the best among all the algorithms in 50 instances and performs the second best in 30 instances. It should be noted that the comparison algorithms such as PDE and BRKGA can find the same best solution as EDA_ACO_LS during the 30 runs, and they outperform the other algorithms in 20 and 18 instances, respectively. In order to explore the stability of the solution results of each algorithm, the distribution of optimization results for 4 classical instances is shown in Fig. 11. From the box graphs, it can be found that the proposed algorithm performs more stable for better results.



(a) The machines and AGVs in the workshop

(b) The layout of the workshop

Fig. 12. The hardware factory.

TABLE VIII
RESULTS OF THE CASE STUDY IN A HARDWARE FACTORY

t/p	EDA_ACO_LS			RGA			PDE			GATS			BRKGA		
	Best(min)	Ave(min)	Div	Best(min)	Ave(min)	Div	Best(min)	Ave(min)	Div	Best(min)	Ave(min)	Div	Best(min)	Ave(min)	Div
0.11	403.7	411.2	7.40e-02	422.4 (−)	432.7	2.27e-01	418.9 (−)	427.9	6.82e-02	411 (−)	426.5	1.57e-01	457 (−)	466.0	1.34e-01

TABLE IX
THE INFORMATION ON THE NEW UPPER BOUNDS

Ins85	Hom_11_44_8_2	Best makespan = 1113.0
job layer:	9 4 6 2 0 3 5 1 4 4 7 2 10 3 6 5 9 4 1 10 3 8 2 6 2 9 1 7 9 5 8 10 1 3 0 0 7 10 5 8 8 6 7 0	
machine layer:	3 1 2 0 2 0 1 0 4 5 3 6 1 4 2 2 5 7 1 2 3 1 4 6 7 4 6 5 7 4 2 6 4 5 3 3 6 7 3 4 5 4 6 7	
AGV layer:	1 0 0 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 1	
Ins86	Hom_12_48_8_2	Best makespan = 1233.0
job layer:	2 4 0 7 11 3 2 1 4 8 4 0 3 9 8 5 2 0 4 1 7 8 11 2 6 9 7 1 11 10 0 3 8 7 6 5 6 3 10 9 5 11 10 1 6 9 10 5	
machine layer:	1 0 2 3 1 0 6 0 4 2 5 6 4 3 2 3 4 6 7 1 5 4 2 7 1 5 4 6 4 1 7 3 5 6 2 3 4 5 2 6 4 7 6 7 4 6 5 3	
AGV layer:	1 0 0 1 1 1 1 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 1 0	
Ins88	Hom_10_58_6_2	Best makespan= 147.0
job layer:	5 3 3 5 8 3 1 0 6 8 5 2 7 3 2 6 1 8 0 4 9 9 7 2 1 0 0 9 2 6 4 9 7 8 2 5 4 6 1 5 3 0 7 9 4 8 1 5 0 9 2 6 4 4 1 7 7 3	
machine layer:	5 1 1 5 1 1 4 5 1 5 5 0 1 1 4 4 4 5 5 1 3 4 1 4 4 0 1 4 4 4 1 5 4 5 4 1 3 4 1 1 1 4 5 5 4 1 1 5 4 5 4 4 1 1 1 5 5 2	
AGV layer:	0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0 1	

5) Discussion of Experimental Results: The better performance of the proposed algorithm is mainly attributed to the following reasons:

1) Considering the similarity of FJCSP with FJSP and AGV-routing problem in different cases, a modeling-based method combining EDA and ACO for describing different features of the two problems is innovated;
2) EDA and ACO generate solutions independently and cooperate by sharing elites. The use of the EDA and ACO is dynamically adjusted to make the algorithm adapt to different problems and search stages for global exploration;
3) The local search with a three-layer operator based on critical path method is proposed to improve local exploitation.

D. A Case Study

Supported by the demonstration project we participated in, we cooperated with the hardware factory in the project and completed a case study. Due to the complexity and diversity of hardware workpieces, the flexible intelligent factory covers a series of processes including lathing, planing, milling, cutting, bending, polishing, drilling and spraying. In addition, the factory has introduced AGVs for transporting semi-finished products under the control of centralized dispatching system.

The aerial view of the factory and the layout of the area for our case study are shown in Fig. 12. The demonstration area mainly consists of four functional areas, namely, the processing area, the standby area, the storage area and the transport track area. In the processing area, different kinds of machines are arranged modularly. In the standby area, 5 AGVs with quick response code navigation are initially placed for transporting materials as also shown in Fig. 12. After the materials are processed along the given routes, they will be temporarily stored in the storage area. For the example of machining bend pipe for a chair, a raw pipe will be processed by cutting, bending, punching, and painting in turn, and finally packaged with other types of finished products for sale.

With the data collected from the factory and the AGV website, the case study is carried out practically. The experiment result of the case study is shown in Table VIII. From this table, it can be found that EDA_ACO_LS can find a better solution than the other four algorithms.

V. Conclusion

In this paper, a multi-view modeling-based hybrid algorithm combining EDA with a novel probability model and ACO with a pheromone model is proposed to minimize the makespan of the scheduling problem for machines and AGVs. A novel EDA whose probability model is designed to describe the information of the operation position and the rule selection for scheduling machines and AGVs is proposed. Simultaneously, a joint model with the job-related and the AGV-related pheromone in ACO is proposed to generate new individuals. In addition, a local search based on a critical path is customized to apply to this problem. An adaptive parameter is designed to regulate the use of the two methods to adapt to the varying demands of multi-view modeling in different cases and search stages. The parameter experiment and comparison experiment are carried out and the advantage of the proposed algorithm is proved by the statistical analysis. Three new upper bounds are found by the proposed algorithm. Finally, we conducted a case study in a hardware factory to finish the actual verification. In the future, the dynamic or multi-objective simultaneous scheduling problem needs to be studied.

Appendix

New bounds of three instances including *Ins85*, *Ins86* and *Ins88* are found, and the solutions are listed in Table IX.

References

[1] J.-Q. Li et al., "A hybrid iterated greedy algorithm for a crane transportation flexible job shop problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2153–2170, Jul. 2022.

[2] W. Shao, D. Pi, and Z. Shao, "A Pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1344–1360, Jul. 2019.

[3] Y. Liu, Y. Zhao, K. Li, S. Yu, and S. Li, "Design and application research of a digitized intelligent factory in a discrete manufacturing industry," *Intell. Autom. Soft Comput.*, vol. 26, no. 5, pp. 1081–1096, 2020.

[4] Y. Yang, M. Zhong, Y. Dessouky, and O. Postolache, "An integrated scheduling method for AGV routing in automated container terminals," *Comput. Ind. Eng.*, vol. 126, pp. 482–493, Dec. 2018.

[5] X. Li et al., "Survey of integrated flexible job shop scheduling problems," *Comput. Ind. Eng.*, vol. 174, Dec. 2022, Art. no. 108786.

[6] D. Y. Lee and F. DiCesare, "Integrated scheduling of flexible manufacturing systems employing automated guided vehicles," *IEEE Trans. Ind. Electron.*, vol. 41, no. 6, pp. 602–610, Dec. 1994.

[7] Ü. Bilge and G. Ulusoy, "A time window approach to simultaneous scheduling of machines and material handling system in an FMS," *Oper. Res.*, vol. 43, no. 6, pp. 1058–1070, Dec. 1995.

[8] P. Lacomme, A. Moukrim, and N. Tchernev, "A new lower bound for scheduling of FMS based on AGV material handling," *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 217–222, 2002.

[9] P. Lacomme, A. Moukrim, and N. Tchernev, "Simultaneous job input sequencing and vehicle dispatching in a single-vehicle automated guided vehicle system: A heuristic branch-and-bound approach coupled with a discrete events simulation model," *Int. J. Prod. Res.*, vol. 43, no. 9, pp. 1911–1942, May 2005.

[10] G. Ulusoy, F. Sivrikaya-Şerifoğlu, and Ü. Bilge, "A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles," *Comput. Oper. Res.*, vol. 24, no. 4, pp. 335–351, Apr. 1997.

[11] I. A. Chaudhry, S. Mahmood, and M. Shami, "Simultaneous scheduling of machines and automated guided vehicles in flexible manufacturing systems using genetic algorithms," *J. Central South Univ.*, vol. 18, no. 5, pp. 1473–1486, Oct. 2011.

[12] M. Badakhshian, "Performance optimization of simultaneous machine and automated guided vehicle scheduling using fuzzy logic controller based genetic algorithm," *Int. J. Phys. Sci.*, vol. 7, no. 9, pp. 1461–1471, Feb. 2012.

[13] H. E. Nouri, O. B. Driss, and K. Ghédira, "Hybrid metaheuristics for scheduling of machines and transport robots in job shop environment," *Int. J. Speech Technol.*, vol. 45, no. 3, pp. 808–828, Oct. 2016.

[14] L. Meng, W. Cheng, B. Zhang, W. Zou, W. Fang, and P. Duan, "An improved genetic algorithm for solving the multi-AGV flexible job shop scheduling problem," *Sensors*, vol. 23, no. 8, p. 3815, Apr. 2023.

[15] A. G. Babu, J. Jerald, A. N. Haq, V. M. Luxmi, and T. P. Vigneswaralu, "Scheduling of machines and automated guided vehicles in FMS using differential evolution," *Int. J. Prod. Res.*, vol. 48, no. 16, pp. 4683–4699, Aug. 2010.

[16] T. Nishi, Y. Hiranaka, and I. E. Grossmann, "A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles," *Comput. Oper. Res.*, vol. 38, no. 5, pp. 876–888, May 2011.

[17] Y. Zheng, Y. Xiao, and Y. Seo, "A Tabu search algorithm for simultaneous machine/AGV scheduling problem," *Int. J. Prod. Res.*, vol. 52, no. 19, pp. 5748–5763, Oct. 2014.

[18] X. Wen, Y. Fu, W. Yang, H. Wang, Y. Zhang, and C. Sun, "An effective hybrid algorithm for joint scheduling of machines and AGVs in flexible job shop," *Meas. Control*, May 2023. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/00202940231173750

[19] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian, "An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs," *Comput. Ind. Eng.*, vol. 86, pp. 2–13, Aug. 2015.

[20] J. T. Lin, C.-C. Chiu, Y.-H. Chang, and H. M. Chen, "A hybrid genetic algorithm for simultaneous scheduling of machines and AGVs in FMS," in *Industrial Engineering, Management Science and Applications*, M. Gen, K. J. Kim, X. Huang, and Y. Hiroshi, Eds. Berlin, Heidelberg: Springer, 2015, pp. 277–286.

[21] M. Nageswararao, K. Narayanarao, and G. Rangajanardhana, "Scheduling of machines and automated guided vehicles in FMS using gravitational search algorithm," *Appl. Mech. Mater.*, vol. 867, pp. 307–313, Jul. 2017.

[22] X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo, "Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems," *IEEE Access*, vol. 7, pp. 74909–74924, 2019.

[23] S. M. Homayouni and D. B. M. M. Fontes, "Production and transport scheduling in flexible job shop manufacturing systems," *J. Global Optim.*, vol. 79, no. 2, pp. 463–502, Feb. 2021.

[24] D. Li, L. Wang, J. Cai, K. Ma, and T. Tan, "Research on terminal distance index-based multi-step ant colony optimization for mobile robot path planning," *IEEE Trans. Autom. Sci. Eng.*, Oct. 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9919849

[25] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and C.-Y. Chen, "Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm," *Adv. Eng. Informat.*, vol. 51, Jan. 2022, Art. no. 101536.

[26] Y. Ding, B. Xin, and J. Chen, "A review of recent advances in coordination between unmanned aerial and ground vehicles," *Unmanned Syst.*, vol. 9, no. 2, pp. 97–117, Apr. 2021.

[27] A. Žilinskas, "A review of statistical models for global optimization," *J. Global Optim.*, vol. 2, no. 2, pp. 145–153, Jun. 1992.

[28] S. M. Homayouni, D. B. M. M. Fontes, and J. F. Gonçalves, "A multistart biased random key genetic algorithm for the flexible job shop scheduling problem with transportation," *Int. Trans. Oper. Res.*, vol. 30, no. 2, pp. 688–716, Mar. 2023.

[29] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm Evol. Comput.*, vol. 2, pp. 1–14, Feb. 2012.

[30] S.-Y. Wang and L. Wang, "An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 1, pp. 139–149, Jan. 2016.

[31] J. Hurink and S. Knust, "Tabu search algorithms for job-shop problems with a single transport robot," *Eur. J. Oper. Res.*, vol. 162, no. 1, pp. 99–111, Apr. 2005.

[32] B. S. P. Reddy and C. S. P. Rao, "A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS," *Int. J. Adv. Manuf. Technol.*, vol. 31, nos. 5–6, pp. 602–613, Nov. 2006.

[33] M. V. S. Kumar, R. Janardhana, and C. S. P. Rao, "Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing," *Int. J. Adv. Manuf. Technol.*, vol. 53, nos. 1–4, pp. 339–351, Mar. 2011.

[34] Q. Zhang, H. Manier, and M.-A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1713–1723, Jul. 2012.

[35] L. Deroussi and S. Norre, "Simultaneous scheduling of machines and vehicles for the flexible job shop problem," in *Proc. Int. Conf. Metaheuristics Nature Inspired Comput.*, 2010, pp. 1–2.

[36] D. C. Montgomery, *Design and Analysis of Experiments*. Hoboken, NJ, USA: Wiley, 2019.

[37] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, Mar. 1947.

Bin Xin (Member, IEEE) received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively. He was an Academic Visitor at the Decision and Cognitive Sciences Research Centre, The University of Manchester, from 2011 to 2012. He is currently a Professor at the School of Automation, Beijing Institute of Technology. His current research interests include search and optimization, evolutionary computation, unmanned systems, and multi-agent systems. He serves as an Associate Editor for many journals, such as *Unmanned Systems*, *Advanced Control for Applications*, *Engineering and Industrial Systems*, and the *Journal of Systems Science and Complexity*.

Sai Lu received the B.E. degree from Harbin Engineering University in 2017 and the M.E. degree from the Beijing Institute of Technology in 2019, where he is currently pursuing the Ph.D. degree in control science and engineering. His current interests include multi-agent systems and combinatorial optimization.

Qing Wang received the B.Eng. and Ph.D. degrees from the University of Science and Technology Beijing, Beijing, China, in 2013 and 2018, respectively. She was a Post-Doctoral Researcher with the School of Automation, Beijing Institute of Technology, Beijing, from 2018 to 2020. She is currently working at the Beijing Institute of Technology. Her current research interests include multi-agent systems, nonlinear systems, and distributed optimization.

Fang Deng (Senior Member, IEEE) received the B.E. and Ph.D. degrees in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2009, respectively. He is currently a Professor with the School of Automation, Beijing Institute of Technology. His research interests include intelligent fire control, intelligent information processing, and smart wearable devices.

Xiang Shi received the B.E. degree in automation from Qingdao University, Qingdao, China, in 2015, and the M.E. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2018. He is currently pursuing the Ph.D. degree in control science and engineering with the Beijing Institute of Technology. His current research interests include intelligent warehousing systems, combinatorial optimization, and reinforcement learning.

Jun Cheng (Member, IEEE) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2006. He is currently with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Professor, and the Director of the Laboratory for Human Machine Control. His current research interests include computer vision, robotics, machine intelligence, and control.

Yuhang Kang received the B.S. degree in measurement and control engineering, the M.S. degree in control science and engineering, and the Ph.D. degree in control science and engineering from Naval Aeronautical and Astronautical University, China, in 2011, 2013, and 2017, respectively. Since 2018, he has been with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, where he is currently an Assistant Researcher with the Human Machine Control Laboratory. His research interests include cooperative control of multi-agent systems and consensus problem.