IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 14 May 2023, accepted 8 June 2023, date of publication 13 June 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3285710

RESEARCH ARTICLE

Research on Multi-Objective Scheduling Algorithm of Job Shop Considering Limited Storage and Transportation Capacity

CHENGFENG PENG[1,2], ZHANTAO LI[2], HONGYANG ZHONG[2,3], XIANG LI[1], ANPING LIN[1], AND YONG LIAO[1,2]

[1]School of Physics, Electronics and Electrical Engineering, Xiangnan University, Chenzhou, Hunan 423000, China
[2]School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, Guangdong 510006, China
[3]Electronic Information School, Foshan Polytechnic, Foshan, Guangdong 528137, China

Corresponding author: Yong Liao (liaoyongxnxy@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51975129, Grant 6197020346, and Grant 72202044; in part by the Guangdong Natural Science Foundation under Grant 2022A1515011165, Grant 2022A1515011175, and Grant 2022A1515010991; in part by the Guangzhou Science and Technology Plan Project Funding under Grant 2023A04J0410; in part by the Hunan Natural Science Foundation under Grant 2020JJ4565; in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 20A460 and Grant 22B0812; in part by the Applied Characteristic Disciplines of Electronic Science and Technology of Xiangnan University under Grant XNXY20221210; in part by the Scientific Research Start-Up Fund for High-Level Talents in Xiangnan University; in part by the Chenzhou Low Carbon Intelligent Manufacturing Technology Research; in part by the 2021 Hunan Colleges and Universities Innovation and Entrepreneurship School-Enterprise Cooperation Base (No.74th); and in part by the Chenzhou Science and Technology Development Plan Project under Grant ZDYF2020161.

ABSTRACT As with the continuous improvement of the workshop automation rate and the importance in energy consumption, more and more enterprises not only need to make scheduling decision on production equipment, but also need to consider whether the scheduling of transportation equipment supports scheduling decisions on workshop production. At the same time, because both workshop production scheduling decision and transportation scheduling decision are NP-hard problems, it is necessary to design an efficient algorithm to improve productivity of the workshop. In order to solve this problem, firstly, based on the analysis of the problem structure, production environment and optimization objectives, a ''manufacturing-transportation'' multi-objective joint scheduling optimization mathematical model is established. By converting the energy consumption into the total transportation time objective of the transportation equipment, both total transportation time and makespan are taken as the optimization objectives. Secondly, based on the design idea of memetic algorithm (MA), non-dominated sorting genetic algorithm-II(NSGA-II) is employed as the basis framework of our new developed algorithm. An effective discrete encoding scheme of MO-MA, a new initialization method for initial population and a neighborhood search mechanism based on critical path are incorporated into our new proposed algorithm. Then the parameter design of the algorithm is completed through variance analysis. Finally, the proposed algorithm is compared and analyzed with other algorithms in the dimension of hypervolume and Set Coverage (SC), and advantages of the algorithm in solving this problem are verified.

INDEX TERMS Job shop, multi-objective optimization, manufacturing-transportation, joint scheduling, memetic algorithm.

I. INTRODUCTION

With the increase in manufacturing labor cost and refinement manufacturing requirements for production, new challenges have been brought to the cost control of multi-variety and small-batch manufacturing enterprises with strong labor dependence, like larger household appliances, customized furniture and 3C digital products [1]. The development of enterprises and technological innovations have greatly promoted the automation level of processing,

The associate editor coordinating the review of this manuscript and approving it for publication was Cheng Qian.

transportation and other equipment used in workshop production and manufacturing systems. In recent years, the developments of basic technologies such as IoT sensing technology and information communication technology have provided a reliable basic technical guarantee for the intelligent transformation of workshop. Consequently, the research on the establishment of a ''manufacturing-transportation'' joint scheduling mechanism for the highly automated intelligent workshop with limited production and transportation capacity has received much attention for its practical significance.

The ''manufacturing-transportation'' joint scheduling problem has been carried out by researchers at an early stage, but the research is mostly limited to the small-scale problems due to the weakness of workshop hardware facility [2], [3], [4], [5], [6], [7], [8], [9], [10]: the mathematical models of prior studies would either directly inherit or make simple extension of the nonlinear mixed integer programming model built by Ulusoy [2], [3], the research on the solution algorithm also focused on the traditional meta-heuristic algorithm [2], [3], heuristic algorithm [4], [5], [6], [7], [8] and relaxation optimization algorithm based on upper/lower bound constraints [9], [10]. And Kunst [10] analyzed the complexity of the flow-shop and the job-shop through the mathematical analysis.

For the past few years, with the continuous upgrading of the manufacturing industry, the continuous deepening of research, and the continuous development of computing resources, some new progress has been made in research on the ''manufacturing-transportation'' joint scheduling problem under the flow shop, job shop and flexible job shop environments. And we have organized some of the research results and summarized them into Table 1.

Through the analysis of the references in Table 1, it can be found that: Due to the fixed product process flow and relatively low transportation flexibility in the manufacturing environment of the flow-shop, the research on joint scheduling problems in the flow-shop mainly focuses on the analysis of complexity of the problem [11], [12], and less on the control in the scheduling process [13]. As the important carriers of the flexible manufacturing workshop, the product process complexity in the manufacturing environment of the job-shop [14], [15], [16], [17], [18], [19], [20], [21], [22], [28], [29] and the flexible job-shop [23], [24], [25], [26], [27], [30], [31], [32], [33], [34] with higher process complexity, which also puts forward higher requirements for the control in the joint scheduling process and get many research results.

As the basic component of the flexible manufacturing environment, a considerable amount of research has been conducted in the job-shop environment. The research results are summarized as follows: Hurink [14] proposed local search algorithms for the job-shop problem with a single AGV where appropriate neighborhood structures are defined by using problem-specific properties. Andy [15] took C_{max} as the optimization objective, and a constraint programming (CP) model is constructed to achieve the optimization solution in a short time; In the doctoral dissertation of Hunang [16], the influence of processing time changes in the scheduling environment was analyzed based on the scheduling results and the number of AGVs, and a variety of meta-heuristic algorithms are proposed to optimally solve the scheduling problem; James et al. [17] built a multi-fidelity model for the joint scheduling problem, and the MO2TOS based on K-mean was proposed to solve the problem. Lacomme et al. [18] built a disjunctive graph model for the problem and proposed an upper bound estimation method and a memetic algorithm based on the model, the approximation of the optimal solution of the problem is achieved, in which C_{max} was taken as the objective. Deroussi et al. [19] improved the algorithm by combining the designed neighborhood search system (iterative mechanism and simulated annealing selection mechanism). Chaudhry et al. [20] proposed a spreadsheet based genetic algorithm (GA) and verified that the proposed approach can be also applied to other problems or objective functions without changing the GA routine or the spreadsheet model. Tamer et al. [21] improved the GA by proposing a heuristic coding scheme to realize the optimal solution of the problem. Tabatabaei et al. [22] took the dynamic scheduling environment as the research background and proposes a dynamic decision-making framework through the predefined method of generating the schedule by a heuristic mechanism. As a more complex production scenario than job-shop, flexible job-shop has higher manufacturing flexibility, and have attracted a wide attention from researchers: Karimi et al. [23] proposed two mixed integer programming models, which combined the imperial competition algorithm and the simulated annealing algorithm. The Taguchi method was adopted to complete the parameter design of the algorithm, and the solution performance of the algorithm was verified through comparative experiments. Andy [24] took C_{max} as the optimization goal and built the CP model of the problem based on the Gurobi solver (a commercial solution tool) to realize the optimal solution of the problem. Dalila et al. [25] proposed a novel mixed integer programming model and solved it with the help of Gurobi by using two sets of chained decisions, which verified the solving ability of the model. Chen et al. [26] proposed an improved hybrid particle swarm optimization (PSO) algorithm by combining the competitive learning mechanism and the random restart mechanism to overcome the weakness of the discrete PSO algorithm. Wu et al. [27] discretized multi-objective differential evolution algorithm and embedded hybrid variable neighborhood search into the algorithm, and the performance of algorithm was improved by enhancing the global search ability of the algorithm. Wang et al. [28] proposed an improved ant colony optimization-simulated annealing algorithm based on a multiattribute dispatching rule to solve the multiload AGVs workshop scheduling problem with limited buffer capacity. However, both job-shop and flexible job-shop scheduling problems are typically addressed by researchers with the small-scale environment [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], and optimization

TABLE 1. A literature review of "manufacturing-transportation" joint scheduling in recent years.

Production Environment	Number of AGVs	The number of Objectives	Optimization Objective	Type of Algorithm	Reference
Flow-shop	Single AGV	Single-Objective	C_{max}	Mathematical Programming	[11]
	Fixed transit time	Single-Objective	Weight sum of completion times		[12]
			C_{max}		[13]
Job-shop	Single-AGV	Single-Objective	C_{max}	Meta-Heuristics	[14]、
	Multi-AGVs	Single-Objective	C_{max}	Mathematical Programming	[15]
				Meta--Heuristics	[16]、[17]、[18]、[19]、[20]、[21]
				Heuristics	[22]
	Multi-AGVs	Multi-Objectives	Minimize the earliness and tardiness processing cost, travelling cost, earliness cost, and tardiness cost	Meta--Heuristics	[29]、[30]
Flexible Job-Shop	Fixed transit time	Single-Objective	C_{max}	Meta-Heuristics	[23]
	Multi-AGVs	Single-Objective	C_{max}	Mathematical Programming	[24]、[25]
				Meta-Heuristics	[26]、[27]、[28]
		Multi-Objectives	Minimize total delivery time、Minimize total trans costs、Minimize total energy consumption、C_{max}、Minimize production costs	Meta-Heuristics	[31]、[32]、[33]、[34]、[35]

algorithms are not designed to consider the limited and short time constraints in actual environments where production planners need to quickly develop production schedules.

For joint scheduling problems, researchers have not limited themselves to single-objective optimization environments and have also conducted research on multi-objective optimization environments: Nabovati et al. [29] proposed a minimum delivery deviation scheme for solving joint scheduling problems by Artificial Immune System algorithm. Mahalakshmi et al. [30] established a new joint scheduling model for the job shop joint scheduling, and a multi-objective invasive weeds optimization algorithm is proposed to solve the problem. Lei et al. [31] established an optimization objective by converting the waiting time and makespan into a waiting time for a flexible flow-shop with dynamic transport waiting times (FFSPDW), and the optimal solution of the problem was obtained to through MA. For the real-world manufacturing system for producing back cover of smart phone, Li et al. [32] took the buffer waiting time and the transportation distance of the AGV as the optimization objectives, the solution optimization of the harmony search algorithm was achieved by an effective discrete harmony encoding scheme. Besides, a new initialization method for harmony memory based on opposition-based learning strategy, a dynamic harmony memory considering rate parameter and a local search strategy were also incorporated into the harmony search algorithm. Feng et al. [33] proposed a GA for optimal decision-making by targeting the total delivery time and total transportation cost for the parallel machines batch scheduling problem. Considering the AGVs battery charge, Mousavi et al. [34] took makespan and the number of AGVs as optimization objectives and proposed a GA-PSO algorithm by combining GA and PSO. The performance is verified through the Flexsim simulation platform. Dai et al. [35] established a multi-objective optimization model for the flexible job shop, where energy consumption and makespan were taken as the objectives, and an improved GA was proposed to solve the problem. Weight-sharing methods are adopted to obtain optimal solutions for multiple-objective optimization problems. However, in real production process, the weights of the solutions are not initially determined, but rely on the judgment of schedule personnel in selecting the solutions that better meet the current production objectives.

After the analysis of existing related works, we can know that the "manufacturing-transportation" joint scheduling problem in flexible manufacturing environment have been conducted in job-shop and flexible job-shop. The majority of these studies are single-objective optimization problems with the objective of minimizing C_{max}, average production cycle or total transportation cost, few studies are associated with multi-objective optimization. Most of multi-objective optimization problems adopt the method of weighted processing, and few studies have been conducted to construct multi-objective optimization algorithms to explore the Pareto frontier solution in a way to find multi-objective optimization solutions. However, in the real-world production environments, the scheduling schemes formulated by production manager are often the balanced results of multiple objectives within a limited amount of time, which cannot be converted from multiple objectives to a single objective simply by assigning weights. Then, the motivation of this paper is to find a reasonable way to solve the ‘‘manufacturing-transportation’’ joint scheduling problem in job-shop, and the following two questions are explored through our research work: 1) How to redesign the state-of-the-art algorithms based on the problem characteristics to enhance their exploitation ability in solving multi-objective optimization problems such as joint scheduling? 2) How to ensure the high quality of scheduling solutions generated by traditional swarm intelligence search algorithms with limited time consumption while ensuring their efficiency in solving large-scale problems?

Based on an analysis of current research and the motivations, this paper takes C_{max} and total transit time as the optimization objectives to study the ‘‘manufacturing-transportation’’ joint scheduling problem. According to the characteristics of the problem, an improve MA algorithm is proposed to solve the multi-objective optimization problem. And to improve the performance of the algorithm, a two-layer coding strategy based on job and AGV, a mixed population initialization method, a crossover and mutation operator, and an efficient neighborhood search strategy based on critical path are designed. Finally, simulation experiments are conducted to verify the superiority of the algorithm.

II. THEOETICAL BACKGROUND

A. BACKGROUND OF THE PROBLEM

A production workshop with an automatic material storage and transportation system is presented in Fig.1. The workshop consists of three areas: a processing area, a transit warehouse and a AGV parking lot. Among them, the processing area is composed of multiple processing units with different functions and rail transportation network, and processing area is responsible for providing processing services for materials or parts. As a material distribution center for materials, parts and semi-finished products, the transit warehouse is responsible for providing material distribution for the workshop. The AGV parking lot is used as a storage point for idle AGV to reduce the probability of workshop blockage. The production process of the jobs in this workshop can be described as: Based on the product process path constraints, the AGV transports the job to the designated processing unit in the processing area to perform the processing operation. After the completion of corresponding processing service, the AGV completes the transfer operation of the job (Transport to the next processing unit required by the job process, or transport to the transit warehouse to perform the storage operation).

B. PROBLEM DESCRIPTION

The ‘‘manufacturing-transportation’’ joint scheduling problem with limited transport capacity can be described as: There is a set of jobs (denoted as J, and $J = \{1, 2, \ldots, i, j, \ldots, n)$ to be processed, and the production process path of job i is O_i ($O_i = \{O_{i1}, O_{i2}, \ldots, O_{ij}, \ldots, O_{in}, O_{i(n+1)}$), the operation O_{ij} can only be processed by the machine M_l ($M_l \in M = \{M_1, M_2, \ldots, M_q, \ldots, M_m$). The transfer service of each job between each loading and unloading node is provided by the set of AGVs (denoted as R, and $R = \{r_1, r_2, \ldots, r_s, \ldots, r_k$). The purpose of our research is to decide the sequence of processing tasks and handling tasks, to achieve the optimization of the scheduling in C_{max} and the total transportation time.

For the convenience of research, the following assumptions are made on the problem:

(1) At the initial moment (which can be regarded as the decision moment), all machines and AGVs are available.

(2) The routes of AGVs between each node are based on the shortest path, and each AGV does not interfere with each other during the process of handling tasks.

(3) The AGV accepts the job from the buffer of the previous process and transports the job to the buffer next to the process.

(4) The time consumption of AGV loading and unloading operations in the buffer area is ignored.

(5) The transportation time of AGVs between each node is only related to the transportation distance and speed of AGVs.

(6) The handling distance between any nodes satisfies the triangle inequality principle, that is, direct transportation will produce less consumption than transit transportation.

(7) At any time, the handling operation can only be carried out by one AGV without interruption, and the failure of the AGV is not considered.

(8) At any time, the processing operation can only be carried out by one machine without interruption, and the failure of the machine is not considered.

(9) There are no associated constraints between different jobs, but for each job, processing operations need to be performed according to the production process path.

(10) The sequence of jobs to be processed follows the first come first serve (FCFS) rule.

(11) All jobs to be processed are initially located in the transit warehouse.

C. MATHEMATICAL MODELS

To facilitate the understanding of the model, the variables and their corresponding meanings are explained as follows:

- **Parameter Setting**

$$\text{Indexes of jobs:} i, l \tag{1}$$

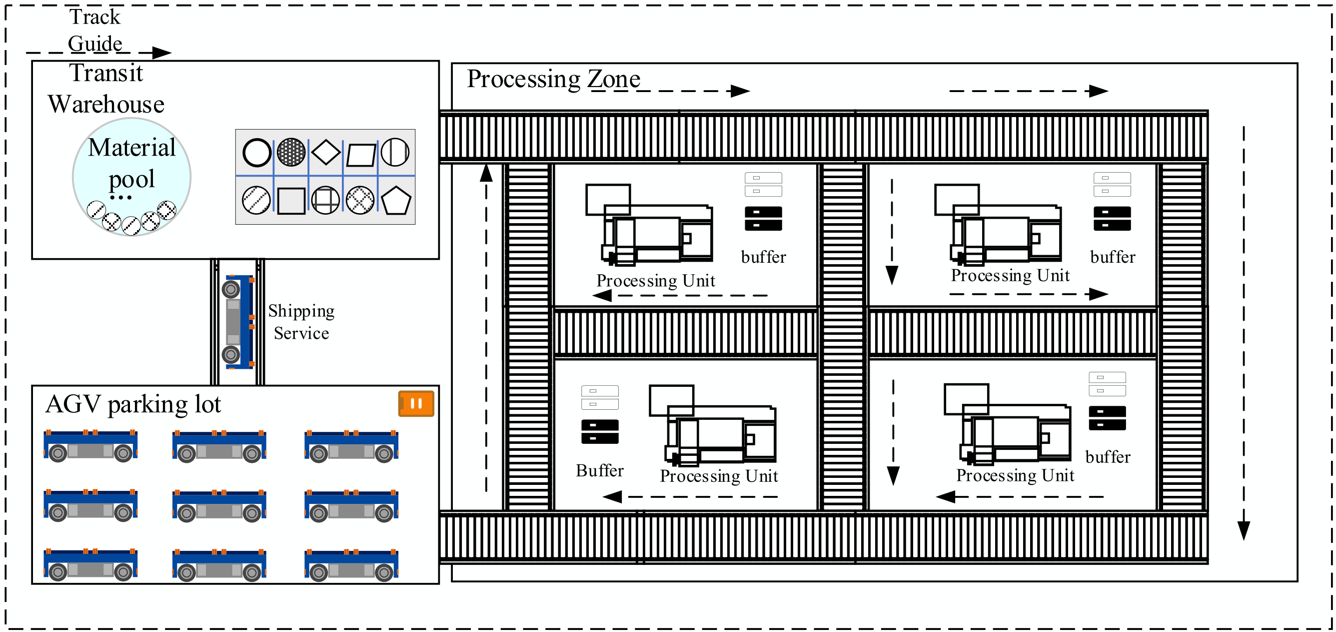


FIGURE 1. Schematic diagram of job shop with limited transportation capacity.

Indexes of machines:M_u (2)

Indexes of the transit warehouse:M_0 (3)

Indexes of AGVs:r_s, r_k (4)

Indexes of operation tasks:O_{ij}, O_{lq} (5)

The release task for job i : O_{i0} (6)

The recycling task for job i : $O_{i(n+1)}$ (7)

The processing time of operation O_{ij} : p_{ij} (8)

The time for load transportation from machine M_q to machine M_u : C_{qu} (9)

The time for no-load transportation from machine M_q to machine M_u : V_{qu}

A very large number:H (10)

- **Variable Setting**

J : a set of jobs to be processed, and

$$J = \{1, 2, \ldots, i, j, \ldots, n \tag{11}$$

M : a set of processing machine,

$$M = \{M_1, M_2, \ldots, M_q, \ldots, M_m \tag{12}$$

R : a set of transportation machine, and

$$R = \{r_1, r_2, \ldots, r_s, \ldots, r_k \tag{13}$$

O_i : the production process path of job$i$, and

$$O_i = \{O_{i1}, O_{i2}, \ldots, O_{ij}, \ldots, O_{in}, O_{i(n+1)} \tag{14}$$

T_{ij} : the transportation task that transports job i from the processing machine of operation O_{ij} to the processing machine of operation$O_{i(j+1)}$ (15)

d_{ij} : the start processing time of operationO_{ij} (16)

f_{ij} : the completed time of operationO_{ij} (17)

d'_{ij} : the start processing time of transportation taskT_{ij} (18)

f'_{ij} : the completed time of transportation taskT_{ij} (19)

- **Decision Variable Setting**

$\alpha^{M_u}_{ij,lq}$: 1, if transportation O_{ij} is processed on machine M_u before operation O_{lq}; 0, otherwise. (20)

β_{ij,r_s} : 1, if transportation task T_{ij} is handled by AGVr_s; 0, otherwise. (21)

$\delta_{ij,lq}$: 1, if transportation task T_{ij} is handled before transportation task T_{lq}; 0, otherwise. (22)

$\lambda_{ij,lq}$: 1, if transportation task T_{ij} and transportation task T_{lq} are both handled by the same AGV; 0, otherwise. (23)

$\theta^{r_s}_{ij,lq}$: 1, if transportation task T_{ij} and transportation task T_{lq} are both handled by the same AGV r_s; 0, otherwise. (24)

- **Objective Optimization Function**

$$f_1 = minC_{max} \tag{25}$$

$$f_2 = \min(\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} \sum\nolimits_{s=1}^{k} \beta_{ij,r_s} \times (C^{r_s}_{M_{ij},M_{i(j+1)}} + V^{r_s}_{M_{ij},M_{i(j+1)}})) \tag{26}$$

The Eq. (25) and (26) are the definitions of two optimization objectives in this problem. Among them, Eq. (25) represents minimizing the C_{max}, and Eq. (26) represents minimizing the total transit time.

- **Subject To**

$$C_{max} \geq f_{i(n+1)} \tag{27}$$

$$f_{ij} \geq d_{ij} + p_{ij} \tag{28}$$
$$p_{i0} = 0, p_{i(m+1)} = 0 \tag{29}$$
$$d_{i(j+1)} \geq f'_{ij} \tag{30}$$
$$d_{i(j+1)} \leq d'_{ij} + \sum_{s=1}^{k} \beta_{ij,r_s} C^{r_s}_{M_{ij},M_{i(j+1)}} \tag{31}$$
$$d'_{ij} \geq f_{ij} \tag{32}$$
$$d_{ij} \geq f_{lq} - H * \alpha^{M_u}_{ij,lq} \tag{33}$$
$$d_{lq} \geq f_{ij} - H * (1 - \alpha^{M_u}_{ij,lq}) \tag{34}$$
$$\alpha^{M_u}_{ij,lq} + \alpha^{M_u}_{lq,ij} \leq 1 \tag{35}$$
$$\sum_{s=1}^{k} \beta_{ij,r_s} = 1 \tag{36}$$
$$\theta^{r_s}_{ij,lq} \geq 1 - \left(1 - \beta_{ij,r_s}\right) * H - \left(1 - \beta_{lq,r_s}\right) * H \tag{37}$$
$$\theta^{r_s}_{ij,lq} \leq \beta_{ij,r_s}, \theta^{r_s}_{ij,lq} \leq \beta_{lq,r_s} \tag{38}$$
$$\lambda_{ij,lq} = \sum_{s=1}^{k} \theta^{r_s}_{ij,lq} \tag{39}$$
$$d'_{ij} \geq d'_{lq} + \sum_{s=1}^{k} \beta_{lq,r_s} \left(C_{M_{lq},M_{l(q+1)}} + V_{M_{l(q+1)},M_{ij}}\right) + \left(\lambda_{ij,lq} - 1\right) * H - \delta_{ij,lq} * H \tag{40}$$
$$d'_{lq} \geq d'_{ij} + \sum_{s=1}^{k} \beta_{ij,r_s} \left(C_{M_{ij},M_{i(j+1)}} + V_{M_{i(j+1)},M_{lq}}\right) + \left(\lambda_{ij,lq} - 1\right) * H - \delta_{ij,lq} * H \tag{41}$$
$$\delta_{ij,lq} + \delta_{lq,ij} = 1 \tag{42}$$
$$C_{pl} = V_{pl} \tag{43}$$

Eq. (27) indicates that C_{max} is the maximum time for all jobs returned to the transit warehouse after processing. Eq. (28) and (29) indicate that once the job is in processing state, it cannot be interrupted. Eq. (30) indicates the update method of the completion time of the job, and Eq. (31) means that there is no processing operation for the job in the transit warehouse. Eq. (32)-(34) represent the transformation relationship between the job processing state and the transportation state in the time dimension: the handling operation can be performed after the processing is completed, and the processing operation can be performed after the handling is completed. Eq. (35)-(37) mean that the processing machine is unique in processing jobs: the machine can only process one job in any time, and the job also can only be processed by one machine in any time. Eq. (38)-(42) represent the uniqueness of performing handling tasks: one AGV can only handle one handling task at the same time, and one handling task can only be served by one AGV. Eq. (43) indicates that the transit time of an unloaded AGV is not discriminated from that of a loaded AGV.

D. PROBLEM ANALYSIS

Based on the analysis framework of KU [36] for the traditional job shop schedule problem, the following characteristics are found by analyzing the problems and mathematical models studied in this joint scheduling problem:

(1) By analyzing the scale of the model and decision variables, it can be seen that the maximum magnitude of the constraint variable is mainly confined by Eq. (37)-(38), and the order of magnitude is $O(k(nm)^2)$. The magnitude of the decision variable is determined by the decision variable $\theta^{r_s}_{ij,lq}$ which is also $O(k(nm)^2)$. Based on the previous analysis, we can plot the change in the number of constraints and the number of decisions as the problem size increases in Fig.2. By the Fig. 2 we can know that when the scale of the problem is 10∗10∗3 (10 machine, 10 jobs and 3 AGV), the decision variables and constraint variables reach the level of 40,000 and 100,000 respectively. When the problem reaches this scale (It is still a small-scales scenario in the actual production environment), it is difficult to obtain a high-quality solution for the established mixed integer programming model in an acceptable time by traditional mathematical programming methods.

(2) The analysis of Eq. (26) shows that the total transportation time consists of two parts: the loaded transportation time and no-loaded transportation time. Among them, the AGVs can be regarded as parallel machines in this paper, and the route of the transportation task is determined, and the execution process of each task is independent and undisturbed, so the loaded transportation time is not affected by the choice of AGVs. The optimization of the total transportation time can be regarded as optimizing the un-load transportation, it can be obtained by the Theorem 1 that when the total transportation time is optimized to a certain range(when he AGV completes a task of transporting and stays in place to wait for further instructions, it will obtain a transportation time plan that is better than other delivery tasks), further optimization will lead to the deterioration of another optimization objective C_{max}.

Theorem 1: If the AGV performs another job handling operation immediately after completing the load transportation task in the schedule P, the objective f_2 of the schedule P will be inferior to the schedule P' in which the AGV waits for the job to be completed by the machine and continues the job handling operation.

Prove: The total transportation time is the cumulative sum of the AGVs' loaded transportation time $\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{s=1}^{k} \beta_{ij,r_s} C^{r_s}_{M_{ij},M_{i(j+1)}}$ and no-load transportation time $\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{s=1}^{k} \beta_{ij,r_s} V^{r_s}_{M_{ij},M_{i(j+1)}}$, and by Eq. (43) we can know that the transportation time between nodes has nothing to do with loading state of the trolley. The route of a job is known and unique, so the load transportation time between schedule P and P' does not make any changes. To optimize the total transportation time, it is necessary to reduce the empty transportation time. The empty transportation time in the schedule P' is 0 (the waiting time is not included in the empty transportation time), but the empty transportation time in the schedule P is a value C greater than 0. Therefore, it is easy to know that $P(f_2) \geq P'(f_2)$.

Based on the above analysis, the pareto optimization can be performed for the two objectives of total transportation time and C_{max}, but it is difficult to solve the problem by

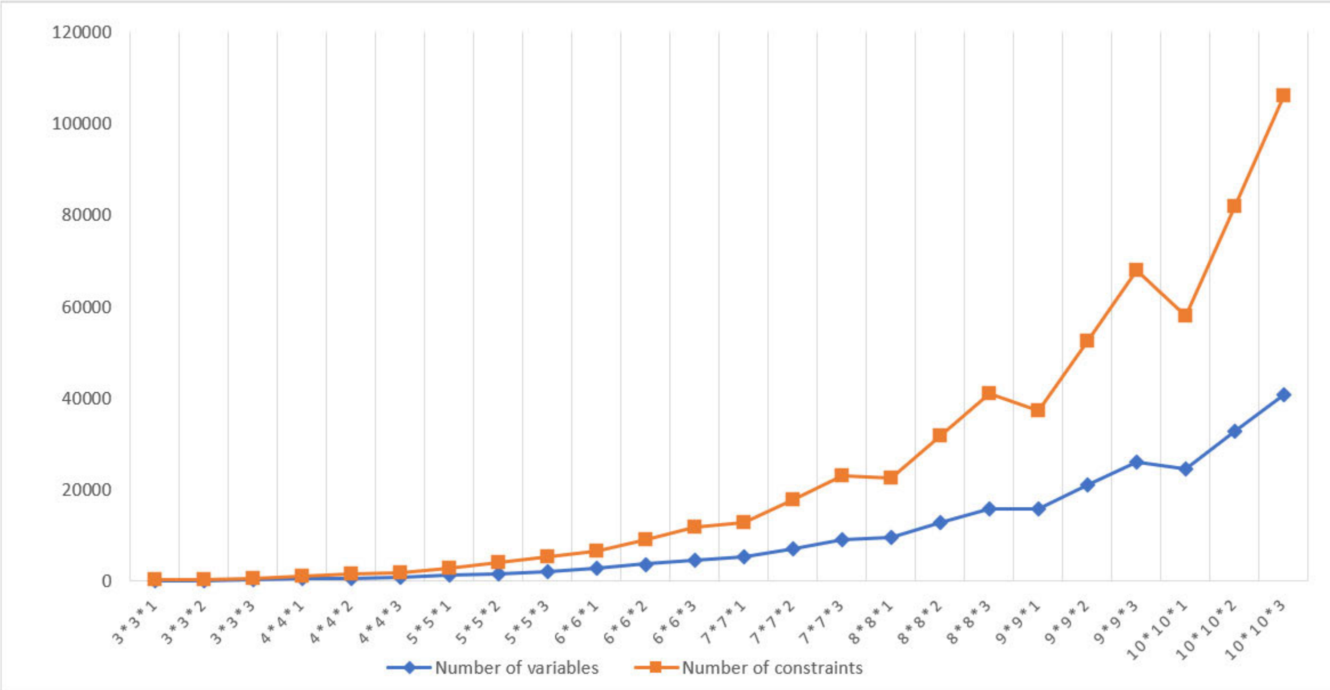


FIGURE 2. The number of constraints and decisions change as the problem size.

the by the exact algorithm. Therefore, we need to propose a meta-heuristic algorithm to solve the multi-objective optimization problem.

III. DESIGN OF MO-MA FOR SOLVING MO-JSPMH

A. THE FRAMEWORK OF MO-MA

The MA is a neighborhood search algorithm that combines the Evolutionary Algorithm (EA) framework with the problem to achieve a balance between exploration and exploitation capabilities in the search process of algorithm [37]. As shown in TABLE 2, we proposed a Multi-objective Memetic Algorithm (MO-MA) based on the design idea of the Memetic Algorithm (MA) for the optimal solution of the MO-JSPMH.

B. ENCODING AND DECODING DESIGN

As shown in Tab. 3 and 4, they are the handling time matrix table and job processing process path information table in the benchmark test cases proposed by Bilge [2]. Among them, Tab. 3 represents the transportation time between any two nodes in the form of a matrix, and Tab. 4 shows the process constraints of the jobs and the processing time for the jobs. The cases shown in Tab. 3 and 4 are taken as examples to describe the encoding and decoding of the MO-MA.

1) GENETIC CODING

Considering that the essence of MO-JSPMH is to make decisions on machine and AGV, the implicit expression of the processing sequence of jobs on the machine can be realized by "prioritizing handling and processing first": the job which is assigned to AGV first has higher priority to be processed in

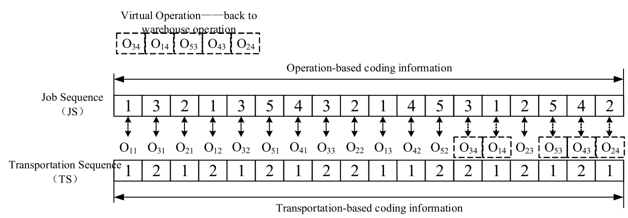


FIGURE 3. The coding patterns of two chain chromosomes.

a machine. Therefore, this study is conducted on the basis of the job coding [38] and designs the coding from the perspective of transportation. As shown in Figure. 3, the encoding consists of two parts: The first part represents the sequence of processing tasks (denoted by JS), the gene value indicates the corresponding job, and the current operation information of the job is indicated by the cumulative frequency of the job from left to right. The second part represents the sequence of transport task (denoted by TS), its code corresponds to the JS, and the gene value represents the AGV which performs the transport task.

2) GENETIC DECODING

After the coding is complete, a corresponding strategy needs to be designed for decoding: in the decoding process, it is necessary to consider the process path constraints of the job as well as the uninterruptibility and exclusivity of the processing and transportation task. Therefore, this paper proposes a decoding method based on the left-shift criterion, and the decoding steps are as follows:

TABLE 2. Transport time of each node.

The operation flow of MO-MA
1: Set the parameters of Population size, Crossover operation, Mutation operation, and Probability of crossover and mutation and maximum number of iterations ($iter_{max}$) et al.
2: Set $iter_{no}=0$ and generate an initial population $P_{(iter_{no})}$ with φ individuals by random and heuristic rules.
3: While $iter_{no} < iter_{max}$ do
4: Generate a new population $P^{c}_{(iter_{no})}$ with φ individuals by crossover operation from $P_{(iter_{no})}$ under probability of crossover
5: Generate a new population $P^{m}_{(iter_{no})}$ with φ individuals by mutation operation form $P^{c}_{(iter_{no})}$ under probability of mutation
6: merge the new population $P^{m}_{(iter_{no})}$ and $P_{(iter_{no})}$ as a new population $P^{child}_{1(iter_{no})}$ with 2φ individuals.
7: Perform deduplication operation on population $P^{child}_{1(iter_{no})}$ (to reduce the computational complexity, two individuals are assumed to be the same if two individuals have the same f_1 and f_2), then get the new population $P^{child}_{11(iter_{no})}$
8: if the population size of $P^{child}_{11(iter_{no})}$ is less than 2φ do
9: generate new individuals by random and heuristic rules to keep the population size as 2φ
10: End if
11: N individuals are selected from population $P^{child}_{11(iter_{no})}$ by random selection strategy
12: A local search is performed on the selected N individuals, then get a new population $P^{child}_{2(iter_{no})}$
13: the individuals in the population $P^{child}_{2(iter_{no})}$ are nondominated sorting and crowding distance
14: $iter_{no} = iter_{no}+1$
15: select φ individuals from the population $P^{child}_{2(iter_{no})}$ by the tournament selection strategy to construct the parent population $P_{(iter_{no})}$
16: End for

TABLE 3. Transport time of each node.

	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	5	8
M3	8	8	6	0	6
M4	6	10	8	6	0

Step 1: Genes are extracted from the JS in order from left to right and converted into corresponding operation, like O_{ip};

Step 2: The processing machine M_{ip} of the operation O_{ip}, the processing machine $M_{i(p-1)}$ of the previous operation $O_{i(p-1)}$ and the processing time of operation are obtained from the production process path information table through the association relationship, and the AGV (denote as r_s) responsible for handling task is obtained through TS gene mapping.

Step 3: The release time $RT^{r_s}_{O_{i(p-1)}}$ of r_s, the released position PO^{r_s} of r_s, and the position $PO^{M_{i(p-1)}}$ of the processing

TABLE 4. Production process path.

	Operation No	Process Time	Machine No
Job 1	O_{11}	8	M1
	O_{12}	16	M2
	O_{13}	12	M4
Job 2	O_{21}	20	M1
	O_{22}	10	M3
	O_{23}	18	M2
Job 3	O_{31}	12	M3
	O_{32}	8	M4
	O_{33}	15	M1
Job 4	O_{41}	14	M4
	O_{42}	18	M3
Job 5	O_{51}	10	M3
	O_{52}	15	M1

machine $M_{i(p-1)}$ are obtained through forward retrieval, then the un-load transportation time $t'_{O_{ip}}$ is calculated, and the time $ST^{r_s}_{O_{i(p-1)}}$ that r_s arrives at the processing machine $M_{i(p-1)}$ is obtained by Eq.(44).

$$ST^{r_s}_{O_{i(p-1)}} = RT^{r_s}_{O_{i(p-1)}} + t'_{O_{ip}} \tag{44}$$

Step 4: The completion time $RT^{r_s}_{O_{i(p-1)}}$ of the operation $O_{i(p-1)}$ in job i is obtained from the job processing information based on the arrival time $ST^{r_s}_{O_{i(p-1)}}$ that obtained in step 3, then the earliest transportation time $RM^{r_s}_{O_{i(p-1)}}$ is got by Eq. (45). The information of position $PO^{M_{i(p-1)}}$ to operation $M_{i(p-1)}$ and position $PO^{M_{ip}}$ to operation M_{ip} is obtained to calculate the load transportation time $t_{O_{ip}}$. Then the time $RT^{r_s}_{O_{ip}}$ that r_s arrive at the processing machine M_{ip} is calculated through Eq. (46), and the AGV's (r_s) released time $RT^{r_s}_{O_{ip}}$ is updated.

$$RM^{r_s}_{O_{i(p-1)}} = \max\{ST^{r_s}_{O_{i(p-1)}}, RT_{O_{i(p-1)}}\} \tag{45}$$

$$RT^{r_s}_{O_{ip}} = RM^{r_s}_{O_{i(p-1)}} + t_{O_{ip}} \tag{46}$$

Step 5: Through the forward search of the machine M_{ip} processing the operation O_{ip}, the release time $RT^{M_{ip}}_{O_{ip}}$ of the machine M_{ip} is obtained after the completion of the predecessor operations. Then the earliest start time $ST^{M_{ip}}_{O_{ip}}$ is obtained by Eq. (47), and the release time $RT_{O_{ip}}$ of job i and the release time of machine M_{ip} are updated.

$$ST^{M_{ip}}_{O_{ip}} = \max\{RT^{M_{ip}}_{O_{ip}}, RT^{r_s}_{O_{ip}}\} \tag{47}$$

Step 6: If all genes in the chromosome have been decoded, the decoding procedure is terminated; otherwise, add 1 to the gene sequence and return to step 1.

As shown in Fig. 4, the decoding results of the cases in Tab. 2 and 3 are obtained under the coding schedule of Fig. 3.

C. INITIAL SOLUTION PRODUCTION STRATEGY

By constructing a high-quality initial solution, the search efficiency of the MO-MA can be effectively improved, and the

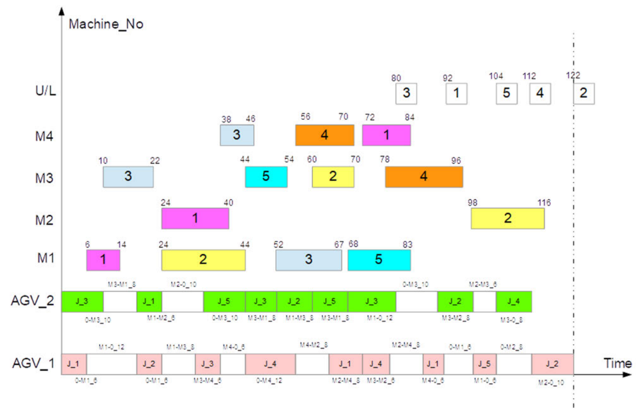


FIGURE 4. A sample of decoding results for MO-JSPMH.

optimization time can be reduced. Especially in the scenario that the search time is limited, a gool initial solution generation mechanism is the basic guarantee of the algorithm's solution quality [37]. Therefore, this paper aims to improve the generation quality of the initial solution by embedding the dispatching rule (DR), and the two design criteria of the DR are as follows:

(1) As shown in Tab. 5, the DR design framework is constructed based on the dynamic transformation relationship between the job state and the resource (like machine and AGV) state. It can be found that the decision result can be described as an AGV performing the handling operation for a job processing by deconstructing the job production process, Hence, the decision can be deconstructed into AGV selection decision and job selection decision, and two strategies are formulated respectively. Then, four initial solution generation methods are generated as follows:

1) Rule1: Combining the earliest handling completion strategy and the earliest operation start strategy.

2) Rule2: Combining the minimum unload transportation time strategy and the earliest operation start strategy.

3) Rule3: Combining the minimum unload transportation time strategy and the earliest operation finish strategy.

4) Rule4: Combining the earliest handling completion strategy and the earliest operation finish strategy.

(2) As shown in Tab. 6, a heuristic rule (denoted as MNEH) considering material transportation is designed based on the NEH [39].

D. CROSSOVER OPERATOR DESIGN

In this paper, a variety of different crossover operators are designed and improved according to the characteristics of the problem, such as AP (Alternating Position Crossover Operator), CX (Cycle Crossover) and SBOX (Similar Block Order Crossover), etc. Based on the results of factor effect analysis, the SBOX with the best comprehensive performance is selected as the crossover operator of the MO-MA. The specific analysis process is referred to the key parameters design part of the MO-MA in the following section.

Because the process difference between the crossover operators is small, the crossover operator of SBOX is used as the representative to illustrate the process of crossover operation, as shown in Fig. 5:

Step 1: Two individuals P_1 and P_2 are randomly selected from the parent population as the starting point for performing the crossover operation.

Step 2: Based on the mapping relationship between the JS and TS, position matching is performed, and the index positions of the parent chromosomes P_1 and P_2 at the same position and the same gene fragment are retrieved. These index positions are inherited to the offspring chromosomes C_1 and C_2, respectively, as shown in Fig. 5(a).

Step 3: Taking the chromosome length N as the base, a number between 1 and N is randomly generated as the cross-gene locus N_g, and the genes before the N_g position in the parent chromosomes P_1 and P_2 are directly inherited to the offspring chromosomes C_1 and C_2 respectively. And the positions keep unchanged, as shown in Fig.5(b).

Step 4: Extract the uninherited genes (genes that exist in the corresponding parent but not in the offspring) in the JS of the parent chromosomes P_1 and C_2, P_2 and C_1 to obtain the missing gene sequence. Following the left-to-right orientation, the non-repetitive insertion of genes in the deleted gene sequence is completed as shown in Fig. 5(c).

Step 5: Extract chromosomes C_1 and C_2 as the result of this crossover operation.

E. MUTATION OPERATOR DESIGN

Based on the previous analysis of the scheduling process of the problem studied in this paper, the complete decision is composed of the processing sequence of the job on the machine and the AGV handling sequence (represented by JS and TS respectively). Therefore, to expand the search space, it is necessary to ensure that the current solution can reach any position in the solution space when designing the mutation operation. Based on this principle, the design of the mutation operator is completed as follows:

(1) The process of mutation operation: First, inherit the population after the crossover operation. Then, each individual in the population is determined whether to perform the mutation operation based on the mutation probability: when it is determined that the mutation operation needs to be performed, JS mutation operation and the TS mutation operation are randomly selected to be performed. Finally, the new individuals after the mutation operation and the unmutated individuals are merged into a new population.

(2) The mutation operator design of JS is shown in Fig. 6(a). First, two gene fragments with different job numbers are randomly selected from individuals. Subsequently, a position-swap operation is performed on the gene fragments which are selected. Finally, the offspring individuals are obtained after mutation operation.

(3) The mutation operator design of TS is shown in Fig. 6(b). First, a gene fragment is randomly selected from the individuals to be mutated. Then, select an AGV that is

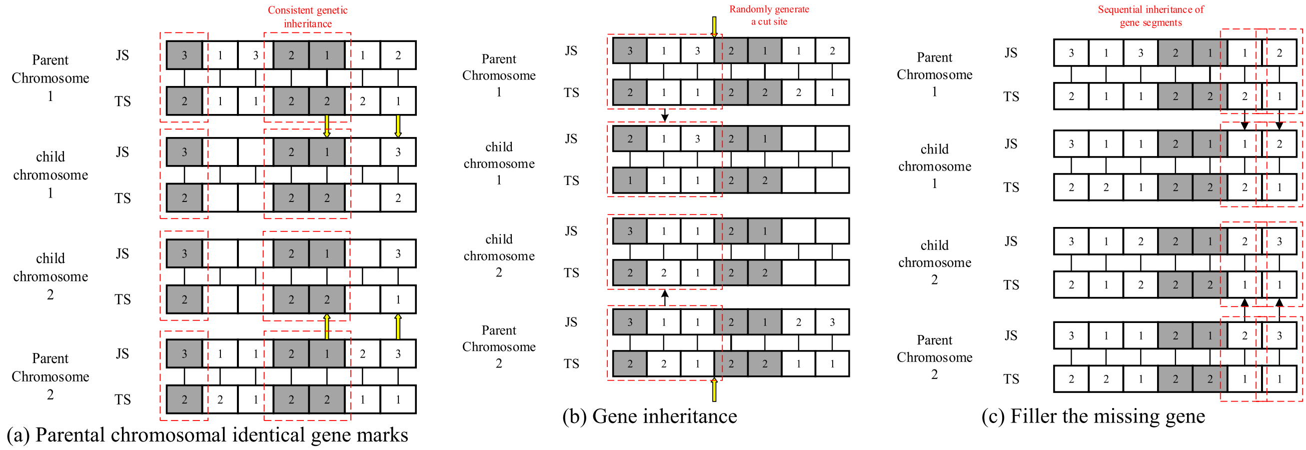


FIGURE 5. The crossover operator of SBOX.

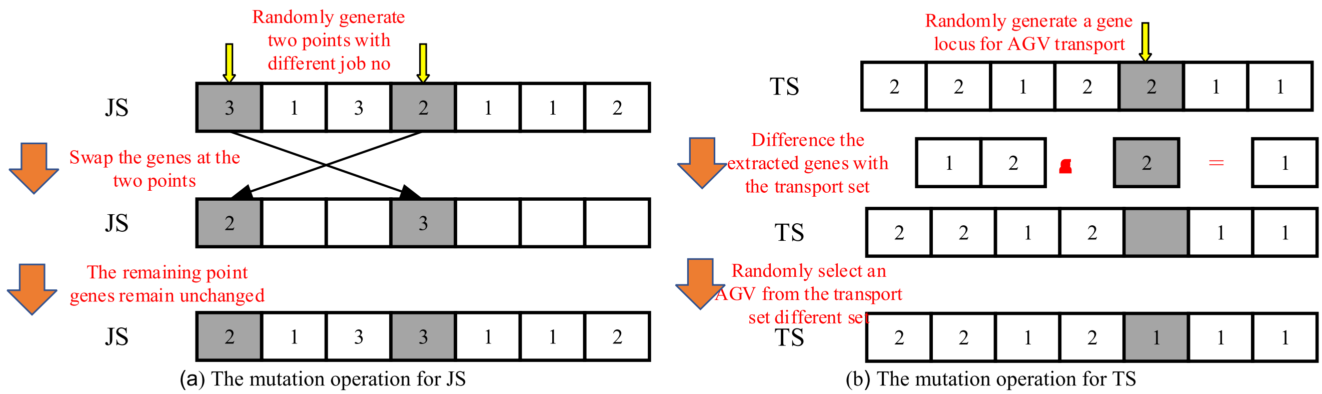


FIGURE 6. The mutation process of SWAP.

different from AGV in the current gene fragment for the replacement operation. Finally, the offspring individuals are got after mutation operation.

F. THE DESIGN OF NEIGHBORHOOD SEARCH OPERATOR

Considering the complexity of the problem, the crossover and mutation operations can only ensure a good exploration ability for algorithm, but the algorithm is still weak in the exploitation ability. As a sub-problem of the MO-JSPMH, job-shop scheduling problem (JSP) is also a classic shop scheduling problem. There are many studies on JSP, and many literatures show that JSP has the following characteristics: For the JSP with C_{max} as optimization objective, the optimization of C_{max} can be achieved quickly by changing the sequence of operations on the critical path. The critical path means that the schedule P is the process path with the longest time from the start node to the completion node, and the critical path is marked as $\mu(P)$ in this paper. The application of critical path theory can still enhance the quality of the schedule of MO-JSPMH by improving the quality of the production scheme:

Theorem 2: For the schedule P, if the operation $O_{ij} \notin \mu(P)$ is moved to obtain a new schedule P', compared with the original schedule P, $C_{max}^{P} \leq C_{max}^{P'}$ holds for the new schedule P'.

Prove: Assuming that $\mu(P)$ is the critical path in the schedule P, the process sequence can be expressed as $Start \to O_1 \to O_2 \to \cdots \to O_l \to End$. Because the moved process O_{ij} does not exist in the original critical path, so it can be divided into two scenarios according to the position to which it is moved.

Scenarios (1): The processed move changes the critical path to $Start \to O_1 \to \cdots \to O_k \to O_{ij} \to O_{k+1} \to \cdots \to O_l \to End$. In this scenario, the number of processes in the critical path increases, then $C_{max}^{P} \leq C_{max}^{P'}$ stands.

Scenarios (2): The processed move does not change the critical path, it means that for the moved process O_{ij}, O_x and O_y are the process before and after the moved position of O_{ij}, respectively. There is no $k \in \{1, 2, \ldots, l-1\}$ for O_x and O_y to make both $O_k = O_x$ and $O_{k+1} = O_y$ hold. And the process information on the critical path has not changed, and $C_{max}^{P} \leq C_{max}^{P'}$ still stands.

Therefore, the next problem to be solved is how to determine the critical path of a schedule in MO-JSPMH. Take O_{ij} as an example to illustrate whether the operator is included in the critical path: determine the earliest start time $S_{O_{ij}}^{Early}$ and latest start time $S_{O_{ij}}^{Last}$ of O_{ij} by Eq. (48)-(50). If $S_{O_{ij}}^{Early} = S_{O_{ij}}^{Last}$

holds for O_{ij}, operation O_{ij} belongs to the critical path.

$$S_{O_{ij}}^{Early} = \max(C_{PO_{ij}}^{Early}, C_{PO_{ij}^k}^{Early}) \tag{48}$$

$$C_{O_{ij}}^{Last} = \max(S_{SO_{ij}}^{Last}, C_{SO_{ij}^k}^{Last}) \tag{49}$$

$$S_{O_{ij}}^{Last} = C_{O_{ij}}^{Last} - P_{O_{ij}} \tag{50}$$

Among them, $C_{PO_{ij}}^{Early}$ is the operation that precedes O_{ij} in job i; $C_{PO_{ij}^k}^{Early}$ is the operation that precedes O_{ij} in machine k; $S_{SO_{ij}}^{Last}$ is the operation that is later processed that O_{ij} in job i, and $C_{SO_{ij}^k}^{Last}$ is the imminently processed operation of O_{ij} in machine k.

After determining the critical path retrieval method of the schedule, the neighborhood search operation can be designed based on the critical path information:

Step 1: An operation (like O_{ij}) is randomly selected from the set of critical path operations.

Step 2: The AGV assignment is adjusted for the selected operation, the new AGV cannot be the original AGV in order to avoid invalid selection.

IV. SIMULATION DESIGN AND ANALYSIS

All simulations in this paper are conducted on the MATLAB R2019b, and the running machine is configured as (Intel(R) Core™i7-8700k CPU@3.70GHZ, 32GB memory). Experiments in any parameter settings are replicated for 30 times independently.

A. CONSTRUCTION THE SIMULATION

Although some researchers have carried out research on the JSPMH, but there are few studies on the MO-JSPMH, and there is a lack of benchmarks to be used. Therefore, to evaluate the effectiveness of the algorithm and the convenience of subsequent research, this paper designs the benchmarks of MO-JSPMH on the basis of the JSPMH [2], [14], [31]. Meanwhile, the scale of the problem is expanded to testify the adaptiveness of the algorithm (The scale of traditional benchmarks is concentrated on five transport nodes, but we expand the test environment of 11 and 16 transport nodes as medium and large-scale benchmark).

The simulations in this paper are carried out based on 2 AGVs with a constant speed of 2m/s, and 50 test cases are designed (40 test cases for small-scale and 10 test cases for medium scale). The details of the test cases are shown in appendix.

B. DESIGN OF EVALUATION INDEX

For multi-objective optimization problems in different optimization environments, existing research includes Set Coverage (SC), Hypervolume, and Inverted Generational Distance (IGD) and other indicators to evaluate the solving performance of the multi-objective optimization algorithm [40]. Considering the particularity of MO-JSPMH, the Hypervolume and the SC are selected to evaluate the performance of the algorithm and testify the effectiveness of the MO-MA in solving the problem.

Among them, the SC index represents the proportion of a solution set B dominated by individuals participating in the comparison solution set A, which can be calculated by Eq. (51) [36], [37], [38], [39], [40], [41]:

$$C(A, B) = \frac{|\{X \epsilon B | \exists y \in A : y \text{ dominates } x\}|}{|B|} \tag{51}$$

In Eq. (51), A and B represents the proportion of individuals in solution set B dominated by individuals in solution set A. The value of SC is a real value from 0 to 1, and the values of $C(A, B)$ and $1 - C(B, A)$ are not necessarily equal. And if $C(A, B) > C(B, A)$ stands, it can be concluded that the solution A is better than the solution B.

Then, the Hypervolume represents the volume of the hypercube constructed by all individuals in the solution set and the reference point in the target space. It is applicable to problem scenarios where the Pareto frontier is unknown: the Hypervolume can be used to measure the convergence, uniformity and generality of the solution set obtained by the algorithm. As an index to evaluate the comprehensiveness of the solution set, the Hypervolume can be calculated by Eq. (52) [37], [38], [39], [40], [41], [42]:

$$H = \sum\nolimits_{i=1}^{NP} |F_2(i) - F_2'| \cdot |F_1(i) - F_1(i-1)| \tag{52}$$

In the Eq. (28-52), H represents the value of Hypervolume, and NP means the number of individuals in the solution set that are non-dominated solutions. $F_1(i)$ and $F_2(i)$ are respectively the objective value of f_1 and f_2 in the ith solution; F_1' and F_2' are divided into the reference values of the reference points in the two target dimensions, and $F_1(0) = F_1'$. When calculating the Hypervolume by Eq. (28), the objective f_1 will be used as the benchmark, and the solution set will be sorted by the value of objective f_1 according to the descending order.

At the same time, the two objectives of C_{max} and total transit time in the MO-JSPMH may have a larger target interval and a smaller target interval in their respective environments with the change of test cases. To avoid the effect of weight bias on the Hypervolume from the original value of any optimization objectives (one of the target values has a large cardinality and occupies the main component of the Hypervolume calculation process), it is necessary to normalize each target value obtained by the operation of each test case by Eq. (53):

$$RDI_{f_i(test_i)} = \frac{f_i(test_j) - f_i(min)}{f_i(max) - f_i(min)} \tag{53}$$

In Eq. (53), $RDI_{f_i(test_i)}$ represents the relative deviation percentage of the schedule $test_i$ in the optimization objective f_i; $f_i(test_i)$ represents the value of the schedule $test_i$ in the optimization objective f_i; and $f_i(min)$ represents the minimum value achieved by all algorithms under the current test case in the objective f_i; $f_i(max)$ represents the maximum value achieved by all algorithms under the current test case in the objective f_i, and the value of RDI ranges between 0 and 1.

TABLE 5. **The pseudocode of dynamic dispatching rules.**

Heuristic dynamic decision algorithm process
1: Initialize AGVs' release time matrix RA, machines' release time matrix RM, jobs' release time matrix RJ
2: Generate the operations' processing information matrix Jobs_OInfo, the total number of operations TO_Num and the number of completed operations of the job Job_Oper
3: for i = 1: TO_Num
4: count the number of jobs with unfinished operations Wait_JobNum
5: for j = 1: Wait_JobNum
6: Determine the position node Job_PointM where the job is waiting to be transported
7: for k = 1:AGV_Num
8: Determine the node AGVR_Point where the current AGV is located, and record the no-load transportation time and transportation completion time of the AGV
9: End for
10: Call **[AGV selection strategy]** to determine the AGV selected to perform the task
11: Determine the target node that the job needs to be transported, and calculate the earliest start time of the operation Job_Earilest completion time of the process Job_FinishT
12: End for
13: Call **[job selection strategy]** to determine the current job task, and update the AGV release time, machine release time, and job release time
14: End for

TABLE 6. **The pseudocode of MNEH.**

MNEH algorithm flow
1: Calculate the total processing time of the jobs to be processed and reorder the jobs according to the total processing time from small to large.
2: Select the jobs in the first and second positions in the reordering sequence and perform full sequence sorting on the jobs: only after the entire process to which the job belongs is completed, the process sorting of another job is performed.
3: When the AGV is called in the sorting, the earliest release strategy of the AGV is called to select the AGV of the operation, the C_{max} obtained by sorting is calculated, and the sorting scheme with the smallest C_{max} among the schemes is selected;
4: for i = 3: Job_SumNum
5: Select the job whose sorting bit is the i index bit, perform the vacancy insertion operation, and use the AGV earliest release strategy in step 3 to select the AGV, and calculate all the insertion position (including 1 pre-position: 0 and n post-position: n+1) C_{max}, choose the sorting scheme with the smallest C_{max}.
6: end for

C. SET THE KEY PARAMETERS OF THE ALGORITHM

In this paper, the full factorial experimental design method [38], [39], [40], [41], [42], [43], which has been widely recognized in the research direction of parameter design, is used to complete the key parameter setting of the algorithm. In the full factorial experimental design, it is necessary to ensure that all relevant factors are subjected to at least one experiment under the corresponding key parameter combination, thus the evaluation of the main experimental parameters can be conducted [38], [39], [40], [41], [42], [43]. As the multi-factor analysis of variance (ANOVA) method is suitable for the significance test of the difference between two or more samples [39], [40], [41], [42], [43], [44], this paper uses the ANOVA method to analyze the experimental results.

TABLE 7. **The result of ANOVA about MO-MA.**

Source	Type III	Df.	Mean sq.	F.	Sig.
P	1.872e7	2	9.360e6	4333.548	.000
CR	1.830e7	2	9.151e6	4236.852	.000
MR	4.139e5	2	2.069e5	95.815	.000
CO	1.336e8	7	1.670e7	7731.901	.000

According to the flow chart of the MO-MA, it is determined that the combination design is carried out from four parameters: crossover operator (CO), population size (P), crossover probability (CR) and mutation probability (MR), and the design of the impact factors of MO-MA is as follows:

crossover operator (CO): AP, CX, OXI, OXII, POS, SB2OX, SBOX, SJ2OX, SJOX.

population size (P): 20, 40, 50.

crossover probability (CR): 0.45, 0.7, 0.95.

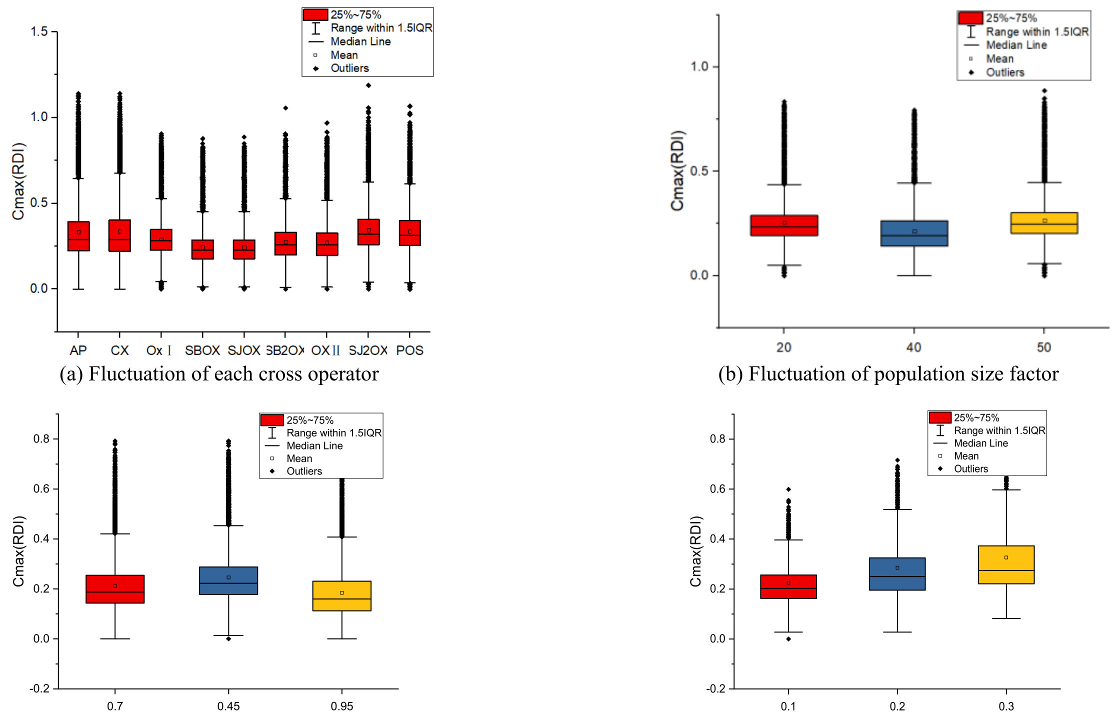
mutation probability (MR): 0.1, 0.2, 0.3.

Considering that this paper is a multi-objective optimization problem, the C_{max} represents the delivery time of the job set, it is a main indicator that receive more attention from the real-world production decision-makers. Therefore, the design of algorithm parameters is carried out with C_{max} as objective: 10 repeated experiments are carried out under each parameter combination environment, and the average value of C_{max} in Pareto solution set of the single experimental result is used as the evaluation standard of current running quality of the algorithm. As shown in Tab. 7, the Sig. corresponding to parameters P, CR, MR and CO are all less than 0.05, indicating that P, CR, MR and CO have significant effects on the algorithm performance under the C_{max} optimization index. Then, it can be seen that the influence of four key parameters on C_{max} are ranked as $CO > P > CR > MR$ by comparing the F-value.

As shown in Fig. 7, according to the influence degree of the factors from strong to weak, the balanced performance of the algorithm under the current level of each factor in the random test case environment is constructed factor by factor.

By analyzing the subgraphs in Fig.7, the key parameters of the MO-MA can be set as follows: the crossover operator is SBOX, the population size is 40, the crossover probability is 0.95, and the mutation probability is 0.1. The reasons are as follows:

(1) As shown in Fig. 7(a), when the crossover operation is SBOX, the performance of the MO-MA is minimally affected by case changes, and its fluctuation interval has a better bias in optimality than other crossover operators.



(a) Fluctuation of each cross operator

(b) Fluctuation of population size factor

(c) Fluctuation of crossover probability factor

(d) Fluctuation of mutation probability factor

FIGURE 7. Statistical box diagram of correlation level values of each factor.

TABLE 8. Hypervolume of MO-MA in different initial solution generation policies(small-scale).

Cases No	Rule1	Rule2	Rule3	Rule4	MNEH
Job_Set1	**1.422444896***	1.328288878	1.33555484	**0.886870885#**	1.194616882
Job_Set2	1.144184175	1.123733163	1.225257909	**1.113554751#**	**1.463504149***
Job_Set3	**1.054954115#**	1.127103452	1.087420993	1.106311214	**1.475178984***
Job_Set4	**1.084063926#**	1.153945782	1.147908623	1.095076673	**1.442482782***
Job_Set5	1.340129837	**1.237437128#**	1.25121989	**1.409033993***	1.248273639
Job_Set6	1.200784156	1.080902339	1.035022567	**0.955469687#**	**1.212695863***
Job_Set7	1.312769316	1.164324999	1.214704353	**0.935312602#**	**1.544049287***
Job_Set8	1.179325539	1.102096035	1.061766388	**0.981469317#**	**1.323493222***
Job_Set9	1.024857362	1.069551445	**1.016297819#**	1.045537327	**1.410712536***
Job_Set10	**0.869414418#**	1.060623133	1.0827393	1.08206391	**1.540225616***
Hypervolume	1.108794126	1.144800635	1.145789268	**1.061070036#**	**1.221445261***

(2) As shown in Fig. 7(b) where the SBOX crossover operator is employed to determine the population size, it can be found that when the population size is 20 or 50, the MO-MA has better performance in terms of volatility, but when the population size is 40, the mean solution has better performance in term of optimality, so the population size is set as 40.

(3) As shown in Fig. 7(c), when the crossover probability is 0.95, the MO-MA outperforms the parameters with crossover probability equal to 0.45 or 0.7 in terms of optimality and stability.

(4) As shown in Fig. 7(d), when the mutation probability is 0.1, the MO-MA has better robustness and quality performance than other parameters.

D. ALGORITHM ANALYSIS

1) ANALYSIS OF THE INFLUENCE OF INITIAL SOLUTION GENERATION STRATEGY ON MO-MA

In order to investigate the impact of different initial solution generation strategies on the MO-MA, this paper incorporates different initial solution generation strategies while keeping the benchmark parameters unchanged, and the Hypervolume is used as the performance evaluation index. By comparing the performance of different initial solution generation strategies in test cases, the influence of the initial solution generation strategies on the MO-MA is analyzed:

(1.2, 1.2) is set as the reference point for Hypervolume calculation [42], on this basis, the value of Hypervolume in 10 experiments of the MO-MA under different initial solution

TABLE 9. Hypervolume of MO-MA in different initial solution generation policies(big-scale).

Initial Generate	Rule1	Rule2	Rule3	Rule4	MNEH
L5_J11	1.161817948	1.212574743	**0.78275365#**	0.97213415	**1.620983404***
L5_J12	0.8677409	**0.972265816***	0.471382981	**0.467542586#**	0.906520727
L5_J13	0.840923741	**1.038273381***	**0.398505266#**	0.518990153	1.004520572
L5_J14	0.851558929	**0.977550372***	0.653786286	**0.419142729#**	0.977506219
L5_J15	0.890139404	0.947184557	0.904358105	**0.425510751#**	**0.978474477***
L6_J16	0.871969507	**0.993154133***	0.434138752	**0.420278495#**	0.967751554
L6_J17	0.899809019	**0.988957881***	**0.301015875#**	0.415494511	0.88733093
L6_J18	0.888396677	**0.979652806***	0.51153911	**0.356841144#**	0.849433727
L6_J19	0.842072256	**1.009496403***	0.851050239	**0.401292017#**	0.944202052
L6_J20	0.889425521	**1.045684587***	0.42541343	**0.349438603#**	0.897163579
Hypervolume	0.90038539	**1.07987965***	0.573394369	**0.474666514#**	0.92930367

TABLE 10. Multiple comparisons test of the different initial solution generation policies.

Hypervolume Value
Tukey HSD

(I)Group	(J)Group	Mean difference(I-J)	Std.error	Sig.	95% confidence interval	
					LB	Upper bound
MNEH	Rule1	0.19839	0.05617	0.004	0.044	0.3527
	Rule2	0.18996	0.05617	0.007	0.0356	0.3443
	Rule3	0.27779	0.05617	0	0.1234	0.4321
	Rule4	0.36531	0.05617	0	0.2109	0.5197
Rule1	MNEH	-0.19839	0.05617	0.004	-0.3527	-0.044
	Rule2	-0.00843	0.05617	1	-0.1628	0.1459
	Rule3	0.0794	0.05617	0.619	-0.075	0.2338
	Rule4	0.16692	0.05617	0.027	0.0126	0.3213
Rule2	MNEH	-0.18996	0.05617	0.007	-0.3443	-0.0356
	Rule1	0.00843	0.05617	1	-0.1459	0.1628
	Rule3	0.08783	0.05617	0.522	-0.0665	0.2422
	Rule4	0.17535	0.05617	0.017	0.021	0.3297
Rule3	MNEH	-0.27779	0.05617	0	-0.4321	-0.1234
	Rule1	-0.0794	0.05617	0.619	-0.2338	0.075
	Rule2	-0.08783	0.05617	0.522	-0.2422	0.0665
	Rule3	0.08752	0.05617	0.526	-0.0668	0.2419
Rule4	MNEH	-0.36531	0.05617	0	-0.5197	-0.2109
	Rule1	-0.16692	0.05617	0.027	-0.3213	-0.0126
	Rule2	-0.17535	0.05617	0.017	-0.3297	-0.021
	Rule3	-0.08752	0.05617	0.526	-0.2419	0.0668

TABLE 11. Homogeneous subsets of the different initial solution generation policies.

Hypervolume Value
Tukey HSD[a]

Group	N	Subset for alpha = 0.05		
		1	2	3
Rule1	50	0.9438		
Rule2	50	1.0313	1.0313	
Rule3	50		1.1107	
Rule4	50		1.1191	
MNEH	50			1.3091
Sig.		0.526	0.522	1

generation strategies in each test environment is calculated. The average value of Hypervolume in 10 experiments is used as the performance of the corresponding initial solution generation strategy in the corresponding case environment. The larger the value, the better the convergence performance of the algorithm.

As shown in Table. 8 and 9, when different initial solution generation strategies are adopted in the small-scale case environment and the large-scale case environment, the average values of Hypervolume in different cases are obtained by MO-MA, respectively. Among them, Table. 8 is the small-scale case test result, and the values in the table represent the average value of Hypervolume under the job set in four different layouts. Table. 9 shows the test result of medium and large-scale cases, and the values in the table represent the average value of Hypervolume under the test cases. Among them, the case identifier "L" represents the

TABLE 12. The compare of MO-MA with other algorithms in Hypervolume (small-scale).

Layouts	Jobs	NSGA-II	MO-PSO	MO-ACO	MO-MA	GAP(Max)
Layout_1	Job_Set1	1.05427346	0.939674044	0.917354606	**1.142796363***	0.197272029
	Job_Set2	0.855472973	0.991621622	0.797972973	**1.181554054***	0.324641162
	Job_Set3	1.063037071	0.915708036	**1.431238775***	1.345330417	0.360198974
	Job_Set4	0.954799383	1.086203704	0.983148148	**1.37087963***	0.120975762
	Job_Set5	1.215873016	0.977048795	1.070711346	1.232169312	0.207049887
	Job_Set6	1.012146878	0.902106674	0.964920933	**1.476976682***	0.389220774
	Job_Set7	1.291825038	0.922855706	1.315414781	**1.429361488***	0.354358073
	Job_Set8	1.044156517	0.898699145	0.942757191	**1.253630474***	0.283122767
	Job_Set9	0.895456053	**1.141903814***	1.110222222	1.102169154	0.193444308
	Job_Set10	0.936258703	0.99811593	1.270324637	**1.35093868***	0.306956921
Layout_2	Job_Set1	1.340468489	0.859988846	0.995025098	**1.421717791***	0.3951058
	Job_Set2	1.043970136	1.200333773	1.187878788	**1.381616162***	0.244384826
	Job_Set3	1.164398169	1.048300103	0.986270861	**1.377409541***	0.283966873
	Job_Set4	1.175511251	1.185598105	1.053324122	**1.331808133***	0.2091022
	Job_Set5	1.451465302	1.272060868	1.148225108	**1.5215427***	0.24535466
	Job_Set6	1.030436727	0.912007366	0.982804525	**1.312454617***	0.305113217
	Job_Set7	1.334927671	0.933959148	0.940105544	**1.433561805***	0.348504442
	Job_Set8	1.277021739	1.12273913	1.086826087	**1.32926087***	0.182383149
	Job_Set9	1.012881099	0.94524319	1.154803855	**1.342678504***	0.296001845
	Job_Set10	1.174519231	1.267036713	1.144326923	**1.455148601***	0.213601331
Layout_3	Job_Set1	1.296500378	1.158650794	1.188412698	**1.436050642***	0.193168569
	Job_Set2	0.899930314	0.923972125	0.96987013	**1.138226164***	0.209357206
	Job_Set3	0.964619165	0.926107406	0.903685504	**1.367539488***	0.339188731
	Job_Set4	1.236857143	1.198666667	1.322285714	**1.42552381***	0.159139498
	Job_Set5	1.15723356	1.035436508	1.136825397	**1.217069161***	0.149237742
	Job_Set6	0.997595105	1.012479383	1.001053472	**1.318903964***	0.243618086
	Job_Set7	1.027939583	0.876700104	0.783998616	**1.229311657***	0.362245846
	Job_Set8	1.714791209	1.086164835	1.187956044	**1.743637363***	0.377069534
	Job_Set9	0.888481781	0.971668916	1.008645974	**1.174165542***	0.243307907
	Job_Set10	0.983203259	0.961033448	0.964085192	**1.486476558***	0.353482271
Layout_4	Job_Set1	1.003579278	1.37950006	0.881226905	**1.007842048***	0.361198357
	Job_Set2	1.098926007	1.014702312	0.98618913	**1.227443822***	0.196550496
	Job_Set3	1.080743007	0.973994755	1.056756993	**1.455708042***	0.330913393
	Job_Set4	1.030856056	1.005428028	0.979403932	**1.10362714***	0.112559037
	Job_Set5	0.911706186	1.173250134	0.93527545	**1.149217329***	0.222922581
	Job_Set6	1.096493716	1.070921689	1.273137286	**1.2946165***	0.172788475
	Job_Set7	1.29476394	1.076292223	1.433932593	**1.503449669***	0.284118221
	Job_Set8	1.294677408	0.961682061	1.117949858	**1.490031252***	0.354589335
	Job_Set9	1.060183955	0.970577414	0.935646398	**1.070904446***	0.12630263
	Job_Set10	1.439082912	1.084347745	1.08861813	**1.459121592***	0.256848949

layout type, and the subsequent values indicate the logistics and transportation matrices used respectively (See the Appendix A and B), while the identifier "J" indicates the job test case, Subsequent values respectively indicate the set of test jobs to be used.

By observing and analyzing the results in Table. 8 and 9, we can see that:

(1) The solution quality of different initial solution generation strategies shows a certain correlation in problems scale:1) In the small-scale test environment, the performance of the MO-MA using MNEH as the initial solution generation strategy is better than other initial solution generation strategies (in all 10 test cases, the optimal solution is obtained in 8 test cases); 2) In the large-scale test environment, the MO-MA solution quality with the Rule2 strategy as the initial solution generation strategy is better than other initial solution generation strategies (in all 10 test cases, the optimal solution is obtained in 8 test cases); 3) With the increase of the problem scale, the MO-MA with Rule3 and Rule4 strategies as the initial solution generation strategy shows obvious deterioration in the solution quality: The solution with poor performance of the obtained Hypervolume are clustered

TABLE 13. The compare of MO-MA with other algorithms in Hypervolume (big-scale).

Layouts	Type of Job	NSGA-II	MO-PSO	MO-ACO	MO-MA	GAP(Max)
Layout_5	Job_Set11	0.987927958	0.883699367	1.138357058	**2.230003054***	0.6037228
	Job_Set12	0.49144348	0.389838372	0.563982364	**1.80427132***	0.783935837
	Job_Set13	0.456892453	0.301721867	0.611951553	**2.03024244***	0.851386287
	Job_Set14	0.35305844	0.306748912	0.568094139	**1.778453472***	0.827519293
	Job_Set15	0.470226923	0.377348907	0.630936721	**1.896440065***	0.801022498
Layout_6	Job_Set16	0.50812971	0.321471629	0.591005453	**1.955417288***	0.835599475
	Job_Set17	0.522904032	0.370868806	0.543824767	**1.87100239***	0.801780688
	Job_Set18	0.453071751	0.37080672	0.579375875	**1.807956226***	0.794902822
	Job_Set19	0.511600189	0.326485363	0.50859958	**1.996981487***	0.836510571
	Job_Set20	0.445656847	0.331952139	0.521508029	**2.036773221***	0.83702057

TABLE 14. Multiple comparisons test of the three traditional algorithms.

Hypervolume Value
Tukey HSD

(I)Group	(J)Group	Mean difference(I-J)	Std. Error	Sig.	95% confidence interval	
					LB	Upper bound
MO-MA	NSGA-II	0.59936	0.06230	0.000	0.4379	0.7608
	MO-PSO	0.77751	0.06230	0.000	0.6161	0.9390
	MO-ACO	0.68206	0.06230	0.000	0.5206	0.8435
NSGA-II	MO-MA	-.0.59936	0.06230	0.000	-0.7608	-0.4379
	MO-PSO	0.17815	0.06230	0.024	0.0167	0.3396
	MO-ACO	0.08270	0.06230	0.547	-0.0787	0.2441
MO-PSO	MO-MA	-0.77751	0.06230	0.000	-0.9390	-0.6161
	NSGA-II	-0.17815	0.06230	0.024	-0.3396	-0.0167
	MO-ACO	-0.9545	0.06230	0.420	-0.2569	0.0660
MO-ACO	MO-MA	-0.68206	0.06230	0.000	-0.8435	-0.5206
	NSGA-II	-0.08270	0.06230	0.547	-0.2441	0.0787
	MO-PSO	0.09545	0.06230	0.420	-0.0660	0.2569

to the algorithms using Rule3 and Rule4, and the inferior solutions obtained in the large-scale test cases are MO-MAs using the initial solution generation strategy with Rule3 and Rule4.

(2) Compared with medium and large-scale environments, the MNEH strategy has an obvious advantage in the proportion of optimal solutions in small-scale environments: MNEH achieves the optimal solution in 8 test sets in the small-scale environment regardless of the layout factor, but optimal solution in the large-scale environment is only obtained in 2 test sets. In this regard, based on the NEH, it has a better solution effect when solving the Flow Shop Scheduling. We speculate that the process path similarity between each two jobs in the test job set in the small-scale environment is greater, so the MNEH strategy shows better solution performance in the small-scale environment.

(3) Compared with small-scale environment, in medium and large-scale environments, the performance of MO-MA solution based on MNEH strategy is degraded due to the increase in the average transportation time between transportation nodes and the increase in the diversity of job process path. The initial solution generation strategy (Rule2) that starts the machine as early as possible still shows better solution performance.

TABLE 15. Homogeneous subsets of the different initial solution generation policies.

Hypervolume Value
Tukey HSD[a]

Group	N	Subset for alpha = 0.05		
		1	2	3
MO-MA	50	1.6000		
NSGA-II	50		1.0006	
MO-ACO	50		0.9179	0.9179
MO-PSO	50			0.8225
Sig.		1.000	0.547	0.420

- **Multiple comparisons of the different initial solution generation policies**

To explore the differences between initial solution generation policies with statistical significance, we referred the method

TABLE 16. The compare of MO-MA with other algorithms in SC (small-scale).

Layouts	Type of Job	C (M, N)	C (N, M)	C (M, P)	C (P, M)	C (M, A)	C (A, M)
Layout_1	Job_Set1	0.977419	0.942308	0.970930	0.770968	1	0.283871
	Job_Set2	0.928328	0.793296	0.970414	0.50838	1	0.354749
	Job_Set3	0.951662	0.537849	1	0.139442	0.986667	0.52988
	Job_Set4	1	0.438095	1	0.114286	0.961957	0.228571
	Job_Set5	0.965517	0.962687	0.982249	0.668966	1	0.482759
	Job_Set6	0.970845	0.5	0.989899	0.230769	0.982222	0.235577
	Job_Set7	0.996743	0.375	0.99359	0.386029	0.845411	0.6875
	Job_Set8	0.888087	0.899135	0.963636	0.677233	0.847107	0.636888
	Job_Set9	0.983553	0.847682	1	0.231788	0.990338	0.248344
	Job_Set10	0.987382	0.685259	0.99422	0.155378	0.977273	0.549801
Layout_2	Job_Set1	0.951965	0.859504	1	0.355372	0.927711	0.404959
	Job_Set2	0.941392	0.824786	1	0.273504	0.936275	0.474359
	Job_Set3	0.924528	0.976974	1	0.230263	0.839572	0.460526
	Job_Set4	0.992308	0.587361	0.993939	0.33829	0.995098	0.308550
	Job_Set5	0.987395	0.689840	1	0.513369	0.977011	0.443850
	Job_Set6	0.996479	0.307339	1	0.119266	0.790576	0.513761
	Job_Set7	1	0.216667	1	0.029167	0.882682	0.608333
	Job_Set8	0.996711	0.586592	1	0.581006	0.810526	0.273743
	Job_Set9	0.983974	0.883162	0.981481	0.358974	0.993377	0.25
	Job_Set10	0.996764	0.484848	0.945455	0.886364	0.934272	0.409091
Layout_3	Job_Set1	0.973154	0.744526	1	0.416058	0.934783	0.474453
	Job_Set2	0.981395	0.82	1	0.41	0.994083	0.52
	Job_Set3	0.981132	0.840816	1	0.183673	0.858896	0.481633
	Job_Set4	0.994819	0.341317	1	0.179641	0.972414	0.389222
	Job_Set5	0.953947	0.941176	0.992424	0.582353	0.872727	0.500000
	Job_Set6	0.995434	0.455026	1	0.010582	0.858586	0.534392
	Job_Set7	0.995902	0.083799	1	0	0.746914	0.307263
	Job_Set8	0.943478	0.731801	1	0.409962	0.870968	0.601533
	Job_Set9	0.976744	0.846809	1	0.123404	0.992701	0.280851
	Job_Set10	0.996212	0.646586	1	0	0.974227	0.313253
Layout_4	Job_Set1	0.954839	0.773463	0.976471	0.734628	0.977401	0.478964
	Job_Set2	0.948052	0.708589	0.994652	0.407975	1	0.159509
	Job_Set3	0.975155	0.708197	0.99	0.468852	1	0.118033
	Job_Set4	0.981413	0.816393	1	0.265574	1	0.177049
	Job_Set5	0.992278	0.742424	0.993333	0.646465	0.987578	0.555556
	Job_Set6	0.993355	0.218107	0.993939	0.415638	0.877273	0.452675
	Job_Set7	0.996466	0.364312	1	0.100372	0.989899	0.531599
	Job_Set8	0.969388	0.804348	0.97546	0.467391	0.970339	0.713768
	Job_Set9	0.946309	0.950769	1	0.206154	0.989899	0.301538
	Job_Set10	0.990415	0.334862	1	0	0.951111	0.252294

in [45] and [46] and conducted multiple comparisons of experimental results, which resulted in the following results as shown in Tab. 10 and 11.

According to Tab. 10, there is a significant difference between MNEH and all the other initial solution generation policies, since their p-value (sig.) is less than the significance level $\alpha = 0.05$. Also, Rule3 has no significant difference between all the other initial solution generation policies except MNEH. Significance has been identified by an asterisk ($*$) in the ''Mean Difference'' column.

The ''Homogeneous subsets'' are provided into Table 11. This table provides an alternative method of computing and displaying the post hoc tests which is regarded as more suitable when group sizes are quite different. In this table, groups listed in the same subset are not significantly different. Consequently, Rule1 and Rule2 are not significantly different. Similarly, Rule2, Rule3 and Rule4 are not significantly different.

By comparing the solution performances between the initial solution generation strategies, each initial solution generation strategy may generate some high-quality gene fragments in each test case, thereby improving the exploration ability of MO-MA. At the same time, based on the flow analysis of the initial solution generation strategy, the time

TABLE 17. The compare of MO-MA with other algorithms in SC (big-scale).

Layouts	Type of Job	C (M, N)	C (N, M)	C (M, P)	C (P, M)	C (M, A)	C (A, M)
Layout_5	Job_Set11	1	0	1	0	1	0
	Job_Set12	1	0	1	0	1	0
	Job_Set13	1	0	1	0	1	0
	Job_Set14	1	0	1	0	1	0
	Job_Set15	1	0	1	0	1	0
Layout_6	Job_Set16	1	0	1	0	1	0
	Job_Set17	1	0	1	0	1	0
	Job_Set18	1	0	1	0	1	0
	Job_Set19	1	0	1	0	1	0
	Job_Set20	1	0	1	0	1	0

TABLE 18. Example problem travel distance for four layouts in meters (small-scale).

Layout_1					
F/T	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

Layout-2					
F/T	L/U	M1	M2	M3	M4
L/U	0	4	6	8	6
M1	6	0	2	4	2
M2	8	12	0	2	4
M3	6	10	12	0	2
M4	4	8	10	12	0

Layout-3					
F/T	L/U	M1	M2	M3	M4
L/U	0	2	4	10	12
M1	12	0	2	8	10
M2	10	12	0	6	8
M3	4	6	8	0	2
M4	2	4	6	12	0

Layout-4					
F/T	L/U	M1	M2	M3	M4
L/U	0	4	8	10	14
M1	18	0	4	6	10
M2	20	14	0	8	6
M3	12	8	6	0	6
M4	14	14	12	6	0

complexity of generating an initial solution is $O(k \cdot (k + n) \cdot a)$, where k represents the number of jobs, n represents the number of process nodes, and A represents the number of AGVs. Through the actual test case operation, the generation of the initial solution can be completed within 0.1s even in a medium and large-scale environment. Therefore, based on the principle of high-quality gene inheritance, multiple initial solution generation strategies are selected in parallel as the initial solution generation mechanism of the final MO-MA.

2) COMPARISON OF MO-MA AND TRADITIONAL ALGORITHMS

Through the analysis of the key parameters of the algorithm and the initial solution generation strategy, the key parameters settings of MO-MA and the initial solution generation mechanism can be determined. In order to verify the effectiveness of the MO-MA in solving the MO-JSPMH, the Hypervolume and SC are used as the algorithm evaluation index to compare MO-MA with NSGA-II [47], Multi-Objective Particle Swarm Optimization (MO-PSO) [48], Multi-Objective Ant Colony Optimization (MO-ACO) [49] and other traditional multi-objective optimization algorithms. And, considering that the algorithms used for comparison are unable to directly be applied to problem-solving, in addition to making adaptability adjustments for the algorithm's problem-solving, we also used the same parameter adjustment method as MO-MA (as shown in *'SET THE KEY PARAMETERS OF THE ALGORITHM'*) to adjust the parameter adjustments of all algorithms used for comparison.

- **Comparison and analysis of convergence of MO-MA**

The Hypervolume of the solution sets obtained by the MO-MA, NSGA-II, MO-PSO and MO-ACO is presented in Tab. 12 and 13 in small-scale problems as well as in the test cases of medium and large-scale problems, respectively. By observing and analyzing the results in Tab. 12 and 13, we can see that:

TABLE 19. Job set data with process time used in example problems(small-scale).

JobSet-1	**JobSet-2**
Job1:M1(8); M2(16); M4(12)	Job1:M1(10); M4(18)
Job2:M1(20); M3(10); M2(18)	Job2:M2(10); M4(18)
Job3:M3(12); M4(8); M1(15)	Job3:M1(10); M3(20)
Job4:M4(14); M2(18)	Job4:M2(10); M3(15); M4(12)
Job5:M3(10); M1(15)	Job5:M1(10); M2(15); M4(12)
	Job6:M1(10); M2(15); M3(12)
JobSet-3	**JobSet-4**
Job1:M1(16); M3(15)	Job1:M4(11); M1(10); M2(7)
Job2:M2(18); M4(15)	Job2:M3(12); M2(10); M4(8)
Job3:M1(20); M2(10)	Job3:M2(7); M3(10); M1(9); M3(8)
Job4:M3(15); M4(10)	Job4:M2(7); M4(8); M1(12); M2(6)
Job5:M1(8); M2(10); M3(15); M4(17)	Job5:M1(9); M2(7); M4(8); M2(10); M3(8)
Job6:M2(10); M3(15); M4(8); M1(15)	
JobSet-5	**JobSet-6**
Job 1: M1(6); M2(12); M4(9)	Job1:M1(9); M2(11); M4(7)
Job 2: M1(18); M3(6); M2(15)	Job2:M1(19); M2(20); M4(13)
Job 3: M3(9); M4(3); M1(12)	Job3:M2(14); M3(20); M4(9)
Job 4: M4(6); M2(15)	Job4: M2(14); M3(20); M4(9)
Job 5: M3(3); M1(9)	Job5:M1(11); M3(16); M4(8)
	Job6:M1(10); M3(12); M4(10)
JobSet-7	**JobSet-8**
Job1:M1(6); M4(6)	Job1:M2(12); M3(21); M4(11)
Job2:M2(11); M4(9)	Job2:M2(12); M3(21); M4(11)
Job3:M2(9); M4(7)	Job3:M2(12); M3(21); M4(11)
Job4:M3(16); M4(7)	Job4:M2(12); M3(21); M4(11)
Job5:M1(9); M3(18)	Job5:M1(10); M2(14); M3(18); M4(9)
Job6:M2(13); M3(19); M4(6)	Job6:M1(10); M2(14); M3(18); M4(9)
Job7:M1(10); M2(9); M3(13)	
Job8:M1(11); M2(9); M4(8)	
JobSet-9	**JobSet-10**
Job1:M3(9); M1(12); M2(9); M4(6)	Job1:M1(11); M3(19); M2(16); M4(13)
Job2:M3(16); M2(11); M4(9)	Job2:M2(21); M3(16); M4(14).
Job3:M1(21); M2(18); M4(7)	Job3:M3(8); M2(10); M1(14); M4(9).
Job4:M2(20); M3(22); M4(11)	Job4:M2(13); M3(20); M4(10:
Job5:M3(14); M1(16); M2(13); M4(9)	Job5:M1(9); M3(16); M4(18).
	Job6:M2(19); M1(21); M3(11); M4(15)

(1) Although the NSGA-II is similar to the MO-MA in terms of convergence performance, the MO-MA outperforms the traditional NSGA-II in all test algorithms. Compared with the NSGA-II, the MO-MA optimizes the depth search strategy, MO-MA shows better exploration ability than NSGA-II in the solution process.

(2) In the small-scale test environment, although MO-MA shows better convergence than other algorithms (in all 40 small-scale test cases, 38 of them achieved optimal solutions), there is little difference in the convergence performance between the algorithms (all the test cases are concentrated, the Hypervolume obtained by the optimal and worst algorithms is within 30% of the maximum GAP), since the scale of the problem is relatively small.

(3) In medium and large-scale environments, the NSGA-II, MO-PSO and MO-ACO have greatly deteriorated their

TABLE 20. Example problem travel distance for four layouts in meters (10 machines).

F/T	L/U	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
L/U	0	22	18	6	8	20	7	25	31	12	27
M1	22	0	11	27	30	32	20	10	28	21	7
M2	18	11	0	22	26	22	21	7	18	12	19
M3	6	27	22	0	6	17	13	29	32	12	33
M4	8	30	26	6	0	23	12	34	38	18	34
M5	20	32	22	17	23	0	28	29	20	11	40
M6	7	20	21	13	12	28	0	26	37	18	23
M7	25	10	7	29	34	29	26	0	19	19	17
M8	31	28	18	32	38	20	37	19	0	19	35
M9	12	21	12	12	18	11	18	19	19	0	28
M10	27	7	19	33	34	40	23	17	35	28	0

ability to solve the MO-JSPMH. The reason for this phenomenon is that the MO-MA can provide better deep search capability support for the iterative search process of the algorithm compared with other algorithms, both the design of the neighborhood search action and the initial solution generation strategy reinforce the exploitation ability of MO-MA. Other algorithms take longer times to explore the solution space of the high-quality solutions, resulting in poor search performance of the algorithm.

- **Multiple comparisons of the three employed traditional algorithms**

According to Table. 14, there is a significant difference between MO-MA and all the other employed algorithms, since their p-value (sig.) is less than the significance level $\alpha = 0.05$. Also, MO-ACO has no significant difference with none of the other algorithms except MO-MA. Significance has been identified by an asterisk (∗) in the "Mean Difference" column.

The "Homogeneous subsets" are provided int Table 15. This table provides an alternative method of computing and displaying the post hoc tests which is considered to be more suitable when group sizes are quite different. In this table, groups listed in the same subset are not significantly different. Consequently, NSGA-II and MO-ACO are not significantly different. Similarly, MO-ACO and MO-PSO are not significantly different.

- **Analysis of ensemble coverage performance of MO-MA**

The SC performance of the Pareto optimal solution set obtained by the MO-MA and other algorithms is shown in Tab. 16 and 17 in small-scale problems and medium and large-scale test environments, respectively. (For typesetting purposes, M, N, P and A are respectively used to represent four algorithms such as MO-MA, NSGA-II, MO-PSO and MO-ACO for convenience). Subsequently, observation and analysis of the results in Tab. 16 and 17 shown as that:

(1) In the small-scale test environment, medium and large-scale test environments, the MO-MA has achieved excellent performance under the SC, and the advantages continue to expand with the increase of problem scale.

(2) In the small-scale test environment, the performance of NSGA-II and MO-MA with SC as the evaluation index is similar in nearly half of the test environment, but MO-MA is still better than NSGA-II. However, there are obvious differences in the SC performance between MO-MA and other optimization algorithms in all test cases.

TABLE 21. **Example problem travel distance for four layouts in meters (15 machines).**

F/T	U/L	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
U/L	0	14	11	33	38	31	31	4	18	22	19	29	4	30	2	19
M1	14	0	11	27	30	32	20	10	28	21	7	22	17	34	16	20
M2	11	11	0	22	26	22	21	7	18	12	19	18	15	23	12	10
M3	33	27	22	0	6	17	13	29	32	12	33	6	37	23	34	16
M4	38	30	26	6	0	23	12	34	38	18	34	8	42	29	39	21
M5	31	32	22	17	23	0	28	29	20	11	40	20	34	6	31	12
M6	31	20	21	13	12	28	0	26	37	18	23	7	35	33	33	21
M7	4	10	7	29	34	29	26	0	19	19	17	25	8	28	6	17
M8	18	28	18	32	38	20	37	19	0	19	35	31	19	16	17	16
M9	22	21	12	12	18	11	18	19	19	0	28	12	26	15	23	3
M10	19	7	19	33	34	40	23	17	35	28	0	27	22	42	21	28
M11	29	22	18	6	8	20	7	25	31	12	27	0	33	25	31	15
M12	4	17	15	37	42	34	35	8	19	26	22	33	0	33	3	23
M13	30	34	23	23	29	6	33	28	16	15	42	25	33	0	30	14
M14	2	16	12	34	39	31	33	6	17	23	21	31	3	30	0	20
M15	19	20	10	16	21	12	32	17	16	3	28	15	23	14	20	0

(3) In the large-scale test environment, the MO-MA shows obvious advantages in all 10 test cases: the solutions obtained by the MO-MA dominate the solutions obtained by all other algorithms, while other optimization algorithms hardly find a solution that dominates MO-MA.

V. CONCLUSION AND FUTURE WORK

This paper takes the ''manufacturing-storage and transportation'' multi-objective joint scheduling problem in the job shop as the research object, and this problem considers the limited transportation capacity constraints of the transportation system. By analyzing the transportation and manufacturing process of the workshop, we proposed a job shop scheduling mathematical model with the limited transportation capacity aiming to minimize transportation time and C_{max} to improve the production efficiency and transit efficiency in workshop operations. For the considered problem, the MO-MA algorithm is proposed, which uses the mixed population initialization method and a critical path-based neighborhood search strategy to improve the search ability based on the algorithm framework of NSGA-II. After that, the proposed MO-MA algorithm is compared to the basic NAGA-II algorithm. Furthermore, two traditional multi-objective optimization algorithms (MO-ACO and MO-PSO) are applied to solve the aforementioned problem to make comprehensive comparisons based on different feature scenarios for algorithm adaptability adjustment. The experimental results reveal that the proposed MO-MA algorithm

TABLE 22. Job set data with process time used in example problems(large-scale for Job_Set11).

Job_Set11
Job1: M4(19), M3(27), M2(39), M5(13), M1(25), M9(37), M10(40), M6(54), M8(74), M7(93)
Job2: M3(69), M1(30), M5(1), M4(4), M2(64), M8(71), M6(2), M10(84), M7(31), M9(8)
Job3: M5(79), M4(80), M1(86), M3(55), M2(54), M9(81), M7(72), M8(86), M6(59), M10(75)
Job4: M3(76), M4(15), M2(26), M1(17), M5(30), M9(44), M8(91), M7(83), M6(52), M10(68)
Job5: M5(68), M4(73), M2(87), M1(74), M3(39), M10(98), M6(100), M9(43), M8(17), M7(77)
Job6: M2(63), M1(49), M3(16), M4(55), M5(9), M10(73), M6(61), M9(34), M7(82), M8(46)
Job7: M1(87), M2(71), M5(43), M4(80), M3(39), M8(70), M9(18), M7(41), M10(79), M6(44)
Job8: M5(70), M3(22), M1(73), M4(62), M2(64), M6(25), M9(19), M7(69), M10(41), M8(28)
Job9: M4(16), M1(84), M2(58), M5(7), M3(9), M6(8), M7(10), M8(17), M9(42), M10(65)
Job10: M4(8), M1(10), M2(3), M5(41), M3(3), M8(40), M9(56), M6(53), M10(96), M7(13)
Job11: M5(62), M2(60), M4(64), M3(12), M1(39), M6(2), M8(64), M7(87), M10(21), M9(60)
Job12: M3(66), M2(71), M4(23), M5(75), M1(78), M8(74), M7(35), M10(24), M9(23), M6(50)
Job13: M2(5), M4(92), M5(6), M1(69), M3(80), M8(13), M6(17), M10(89), M7(80), M9(47)
Job14: M1(82), M4(84), M2(24), M3(47), M5(93), M8(85), M6(34), M7(73), M9(28), M10(91)
Job15: M5(55), M1(57), M4(63), M3(24), M2(40), M8(30), M7(37), M6(99), M9(88), M10(41)
Job16: M2(75), M3(47), M4(68), M1(7), M5(78), M8(80), M7(2), M10(23), M9(49), M6(50)
Job17: M1(91), M5(25), M3(10), M2(21), M4(94), M9(6), M8(59), M6(84), M10(75), M7(70)
Job18: M3(85), M2(31), M1(94), M5(94), M4(11), M6(21), M10(7), M7(61), M9(50), M8(93)
Job19: M2(27), M1(77), M5(13), M3(30), M4(2), M6(88), M8(4), M10(39), M7(53), M9(54)
Job20: M2(34), M3(12), M4(31), M1(24), M5(24), M8(16), M6(6), M10(88), M9(81), M7(11)

TABLE 23. Job set data with process time used in example problems (large-scale for Job_Set12).

Job_Set12
Job1:M3(19), M1(30), M2(68), M5(55), M4(24), M9(34), M8(72), M6(32), M10(62), M7(45)
Job2: M3(63), M2(11), M5(65), M4(16), M1(67), M10(95), M9(23), M8(82), M7(52), M6(53)
Job3: M3(19), M5(17), M2(79), M4(49), M1(12), M8(41), M10(67), M9(40), M7(25), M6(42)
Job4: M1(42), M3(71), M4(27), M5(95), M2(19), M6(48), M9(100), M7(31), M8(25), M10(38)
Job5: M4(1), M2(100), M5(68), M1(94), M3(89), M6(86), M8(35), M10(29), M9(56), M7(55)
Job6: M5(93), M2(53), M3(4), M4(48), M1(57), M9(99), M8(67), M6(86), M7(80), M10(60)
Job7: M5(82), M2(95), M3(12), M1(60), M4(80), M9(88), M8(5), M7(81), M10(52), M6(69)
Job8: M4(79), M2(31), M5(63), M1(28), M3(64), M9(63), M6(29), M8(75), M10(18), M7(33)
Job9: M5(9), M2(64), M3(31), M1(13), M4(33), M10(82), M7(79), M6(30), M8(84), M9(20)
Job10: M3(14), M1(56), M2(95), M5(34), M4(13), M7(16), M6(44), M8(45), M9(62), M10(86)
Job11: M5(66), M4(9), M3(66), M2(46), M1(12), M6(10), M8(58), M7(6), M9(62), M10(17)
Job12: M5(89), M2(52), M3(37), M4(74), M1(7), M9(43), M6(96), M8(89), M7(21), M10(66)
Job13: M2(73), M4(68), M3(5), M5(49), M1(67), M10(23), M8(7), M6(44), M9(30), M7(29)
Job14: M3(21), M1(68), M2(88), M5(75), M4(64), M7(6), M9(72), M8(66), M10(66), M6(56)
Job15: M2(24), M5(25), M3(69), M1(27), M4(51), M10(60), M9(26), M7(45), M6(77), M8(93)
Job16: M3(19), M4(17), M2(82), M5(75), M1(34), M6(67), M10(89), M7(91), M8(13), M9(35)
Job17: M5(2), M1(21), M4(83), M2(19), M3(65), M7(65), M9(8), M10(68), M8(60), M6(7)
Job18: M2(63), M4(49), M3(4), M5(2), M1(50), M10(99), M6(27), M7(68), M9(46), M8(89)
Job19: M1(48), M5(45), M4(100), M3(66), M2(30), M7(58), M8(73), M10(94), M6(36), M9(5)
Job20: M3(36), M1(53), M5(56), M4(57), M2(77), M10(7), M7(59), M9(8), M6(15), M8(23)

outperforms the other employed algorithms in this paper. The outcomes of the conducted experiments reveal that:

(1) For scenarios where the solution time is limited, the search efficiency of the MO-MA for the effective solution

TABLE 24. Job set data with process time used in example problems (large-scale for Job_Set13).

Job_Set13
Job1: M3(16), M2(58), M1(22), M5(24), M4(53), M9(9), M10(57), M8(63), M6(92), M7(43)
Job2: M4(6), M2(48), M5(14), M1(66), M3(24), M8(2), M10(85), M7(73), M9(19), M6(99)
Job3: M5(100), M3(90), M1(63), M2(14), M4(31), M6(27), M10(15), M9(1), M7(51), M8(33)
Job4: M3(98), M4(84), M5(52), M1(12), M2(96), M10(60), M7(74), M9(93), M6(45), M8(49)
Job5: M5(39), M1(54), M3(28), M4(8), M2(30), M9(57), M7(75), M6(9), M8(41), M10(19)
Job6: M4(94), M1(8), M3(89), M2(13), M5(37), M9(36), M7(63), M10(24), M6(71), M8(97)
Job7: M4(90), M3(69), M2(25), M5(15), M1(65), M8(52), M7(56), M10(91), M9(83), M6(86)
Job8: M4(59), M2(99), M5(41), M1(68), M3(14), M8(4), M10(55), M7(48), M9(13), M6(15)
Job9: M5(36), M3(17), M2(51), M1(16), M4(54), M9(45), M6(50), M8(98), M7(68), M10(82)
Job10: M2(57), M1(11), M5(55), M3(93), M4(51), M7(61), M10(40), M8(19), M9(24), M6(55)
Job11: M5(56), M1(73), M4(59), M3(38), M2(51), M7(99), M9(29), M10(53), M6(7), M8(72)
Job12: M4(68), M5(50), M2(88), M3(88), M1(33), M6(47), M9(52), M7(26), M10(74), M8(68)
Job13: M3(3), M4(42), M1(45), M2(57), M5(28), M6(14), M9(22), M10(31), M7(44), M8(38)
Job14: M4(89), M1(73), M5(12), M2(9), M3(49), M6(11), M9(15), M8(41), M10(37), M7(10)
Job15: M4(76), M3(97), M5(100), M2(92), M1(25), M6(8), M10(92), M8(51), M7(58), M9(65)
Job16: M5(50), M1(54), M4(85), M2(47), M3(45), M7(99), M10(39), M6(32), M9(87), M8(56)
Job17: M1(70), M3(58), M4(33), M2(85), M5(25), M9(5), M8(65), M10(20), M7(52), M6(44)
Job18: M2(22), M4(45), M5(60), M1(66), M3(5), M8(61), M7(73), M10(60), M6(14), M9(44)
Job19: M5(64), M1(97), M3(31), M2(4), M4(43), M10(47), M8(93), M7(100), M6(10), M9(51)
Job20: M4(9), M5(87), M3(34), M1(62), M2(56), M6(66), M9(95), M8(56), M10(42), M7(86)

TABLE 25. Job set data with process time used in example problems (large-scale for Job_Set14).

Job_Set14
Job1: M3(16), M1(59), M5(10), M4(95), M2(64), M9(92), M10(56), M8(3), M6(73), M7(17)
Job2: M2(5), M5(64), M4(30), M3(14), M1(96), M10(11), M9(73), M8(35), M7(93), M6(12)
Job3: M4(35), M5(75), M1(54), M2(30), M3(83), M10(20), M9(29), M8(38), M7(90), M6(39)
Job4: M5(29), M4(21), M1(52), M3(93), M2(20), M6(5), M8(11), M9(53), M10(56), M7(98)
Job5: M1(17), M4(16), M5(41), M2(78), M3(100), M6(55), M9(27), M7(2), M8(87), M10(55)
Job6: M4(97), M2(32), M5(84), M3(71), M1(38), M10(64), M8(16), M6(5), M7(41), M9(41)
Job7: M4(41), M2(57), M5(37), M1(64), M3(92), M7(19), M10(47), M8(94), M9(79), M6(21)
Job8: M1(23), M4(67), M2(39), M5(98), M3(63), M9(83), M6(45), M7(89), M10(81), M8(44)
Job9: M2(88), M1(59), M4(39), M3(63), M5(91), M9(36), M6(44), M7(45), M10(43), M8(12)
Job10: M3(29), M2(17), M1(6), M4(74), M5(51), M10(14), M7(2), M6(56), M8(49), M9(14)
Job11: M4(75), M3(10), M5(1), M1(35), M2(99), M8(56), M6(95), M10(78), M7(53), M9(82)
Job12: M1(75), M3(96), M2(21), M4(90), M5(55), M7(23), M8(40), M10(76), M9(55), M6(45)
Job13: M4(90), M5(64), M1(72), M3(33), M2(59), M8(51), M7(74), M6(85), M10(76), M9(38)
Job14: M4(57), M2(84), M3(87), M5(2), M1(68), M9(4), M6(77), M7(37), M8(37), M10(94)
Job15: M2(16), M4(46), M5(34), M3(23), M1(77), M8(68), M9(14), M10(54), M6(37), M7(99)
Job16: M5(24), M2(73), M3(92), M1(43), M4(42), M6(81), M8(99), M9(88), M10(80), M7(5)
Job17: M2(56), M3(51), M1(3), M5(87), M4(25), M6(62), M8(11), M9(88), M7(68), M10(29)
Job18: M3(85), M4(3), M5(21), M1(49), M2(79), M9(38), M6(37), M10(72), M8(18), M7(18)
Job19: M1(2), M4(55), M2(31), M3(29), M5(98), M6(92), M7(43), M9(99), M8(67), M10(41)
Job20: M5(69), M4(64), M1(61), M2(13), M3(31), M6(6), M9(84), M10(94), M8(32), M7(54)

space can be improved by embedding an initial solution generation strategy with high solution quality, the search time of MO-MA can be reduced, thus the solution quality of the algorithm can be guaranteed in a limited time. Especially

TABLE 26. Job set data with process time used in example problems (large-scale for Job_Set15).

Job_Set15
Job1: M3(19), M2(30), M4(80), M1(84), M5(14), M9(51), M6(73), M7(91), M8(81), M10(71)
Job2: M3(74), M5(79), M2(39), M1(7), M4(66), M10(6), M6(93), M9(76), M7(21), M8(76)
Job3: M5(90), M4(33), M2(38), M3(73), M1(61), M9(61), M8(76), M6(86), M10(28), M7(35)
Job4: M5(1), M4(22), M3(1), M1(77), M2(33), M7(98), M6(4), M10(27), M9(8), M8(68)
Job5: M3(63), M5(5), M2(95), M1(7), M4(50), M9(46), M10(28), M7(70), M6(60), M8(34)
Job6: M1(98), M2(73), M5(15), M4(21), M3(32), M8(24), M10(9), M9(24), M6(7), M7(34)
Job7: M4(51), M5(47), M3(30), M2(16), M1(51), M6(41), M7(79), M8(79), M10(3), M9(72)
Job8: M5(3), M2(59), M1(53), M4(20), M3(19), M7(20), M10(16), M8(90), M6(96), M9(18)
Job9: M2(34), M3(55), M4(97), M1(93), M5(90), M8(81), M6(63), M9(41), M7(1), M10(51)
Job10: M5(77), M4(87), M2(92), M3(83), M1(45), M8(75), M10(60), M7(75), M6(93), M9(33)
Job11: M1(31), M3(66), M2(58), M5(17), M4(94), M6(63), M8(80), M10(61), M7(78), M9(52)
Job12: M5(70), M2(25), M3(75), M1(89), M4(41), M8(100), M6(73), M7(28), M9(94), M10(88)
Job13: M2(67), M5(62), M4(12), M3(55), M1(62), M6(58), M9(66), M8(73), M7(55), M10(1)
Job14: M5(81), M1(37), M2(2), M4(39), M3(17), M8(74), M7(71), M9(61), M6(42), M10(5)
Job15: M4(62), M1(31), M5(63), M3(31), M2(5), M10(7), M8(77), M9(34), M7(34), M6(3)
Job16: M1(5), M3(55), M4(62), M2(82), M5(80), M7(6), M9(7), M8(29), M6(80), M10(89)
Job17: M4(26), M2(50), M3(58), M1(22), M5(68), M8(12), M7(9), M10(34), M6(90), M9(87)
Job18: M1(50), M3(28), M2(64), M5(34), M4(63), M8(9), M10(48), M7(63), M9(61), M6(2)
Job19: M1(47), M3(23), M2(23), M5(82), M4(98), M8(66), M7(78), M9(100), M10(79), M6(32)
Job20: M2(13), M5(14), M1(90), M3(77), M4(80), M10(30), M8(31), M6(36), M7(51), M9(69)

TABLE 27. Job set data with process time used in example problems (large-scale for Job_Set16).

Job_Set14
Job1: M3(16), M1(59), M5(10), M4(95), M2(64), M9(92), M10(56), M8(3), M6(73), M7(17)
Job2: M2(5), M5(64), M4(30), M3(14), M1(96), M10(11), M9(73), M8(35), M7(93), M6(12)
Job3: M4(35), M5(75), M1(54), M2(30), M3(83), M10(20), M9(29), M8(38), M7(90), M6(39)
Job4: M5(29), M4(21), M1(52), M3(93), M2(20), M6(5), M8(11), M9(53), M10(56), M7(98)
Job5: M1(17), M4(16), M5(41), M2(78), M3(100), M6(55), M9(27), M7(2), M8(87), M10(55)
Job6: M4(97), M2(32), M5(84), M3(71), M1(38), M10(64), M8(16), M6(5), M7(41), M9(41)
Job7: M4(41), M2(57), M5(37), M1(64), M3(92), M7(19), M10(47), M8(94), M9(79), M6(21)
Job8: M1(23), M4(67), M2(39), M5(98), M3(63), M9(83), M6(45), M7(89), M10(81), M8(44)
Job9: M2(88), M1(59), M4(39), M3(63), M5(91), M9(36), M6(44), M7(45), M10(43), M8(12)
Job10: M3(29), M2(17), M1(6), M4(74), M5(51), M10(14), M7(2), M6(56), M8(49), M9(14)
Job11: M4(75), M3(10), M5(1), M1(35), M2(99), M8(56), M6(95), M10(78), M7(53), M9(82)
Job12: M1(75), M3(96), M2(21), M4(90), M5(55), M7(23), M8(40), M10(76), M9(55), M6(45)
Job13: M4(90), M5(64), M1(72), M3(33), M2(59), M8(51), M7(74), M6(85), M10(76), M9(38)
Job14: M4(57), M2(84), M3(87), M5(2), M1(68), M9(4), M6(77), M7(37), M8(37), M10(94)
Job15: M2(16), M4(46), M5(34), M3(23), M1(77), M8(68), M9(14), M10(54), M6(37), M7(99)
Job16: M5(24), M2(73), M3(92), M1(43), M4(42), M6(81), M8(99), M9(88), M10(80), M7(5)
Job17: M2(56), M3(51), M1(3), M5(87), M4(25), M6(62), M8(11), M9(88), M7(68), M10(29)
Job18: M3(85), M4(3), M5(21), M1(49), M2(79), M9(38), M6(37), M10(72), M8(18), M7(18)
Job19: M1(2), M4(55), M2(31), M3(29), M5(98), M6(92), M7(43), M9(99), M8(67), M10(41)
Job20: M5(69), M4(64), M1(61), M2(13), M3(31), M6(6), M9(84), M10(94), M8(32), M7(54)

in large-scale environments, the optimization algorithm with high-quality solution inheritance shows better utilization of search time than the random search strategy.

(2) During the process of intelligent decision-making and factory upgrading, production managers can ensure the quality of production decisions by transforming their

TABLE 28. Job set data with process time used in example problems (large-scale for Job_Set17).

Job_Set15
Job1: M3(19), M2(30), M4(80), M1(84), M5(14), M9(51), M6(73), M7(91), M8(81), M10(71)
Job2: M3(74), M5(79), M2(39), M1(7), M4(66), M10(6), M6(93), M9(76), M7(21), M8(76)
Job3: M5(90), M4(33), M2(38), M3(73), M1(61), M9(61), M8(76), M6(86), M10(28), M7(35)
Job4: M5(1), M4(22), M3(1), M1(77), M2(33), M7(98), M6(4), M10(27), M9(8), M8(68)
Job5: M3(63), M5(5), M2(95), M1(7), M4(50), M9(46), M10(28), M7(70), M6(60), M8(34)
Job6: M1(98), M2(73), M5(15), M4(21), M3(32), M8(24), M10(9), M9(24), M6(7), M7(34)
Job7: M4(51), M5(47), M3(30), M2(16), M1(51), M6(41), M7(79), M8(79), M10(3), M9(72)
Job8: M5(3), M2(59), M1(53), M4(20), M3(19), M7(20), M10(16), M8(90), M6(96), M9(18)
Job9: M2(34), M3(55), M4(97), M1(93), M5(90), M8(81), M6(63), M9(41), M7(1), M10(51)
Job10: M5(77), M4(87), M2(92), M3(83), M1(45), M8(75), M10(60), M7(75), M6(93), M9(33)
Job11: M1(31), M3(66), M2(58), M5(17), M4(94), M6(63), M8(80), M10(61), M7(78), M9(52)
Job12: M5(70), M2(25), M3(75), M1(89), M4(41), M8(100), M6(73), M7(28), M9(94), M10(88)
Job13: M2(67), M5(62), M4(12), M3(55), M1(62), M6(58), M9(66), M8(73), M7(55), M10(1)
Job14: M5(81), M1(37), M2(2), M4(39), M3(17), M8(74), M7(71), M9(61), M6(42), M10(5)
Job15: M4(62), M1(31), M5(63), M3(31), M2(5), M10(7), M8(77), M9(34), M7(34), M6(3)
Job16: M1(5), M3(55), M4(62), M2(82), M5(80), M7(6), M9(7), M8(29), M6(80), M10(89)
Job17: M4(26), M2(50), M3(58), M1(22), M5(68), M8(12), M7(9), M10(34), M6(90), M9(87)
Job18: M1(50), M3(28), M2(64), M5(34), M4(63), M8(9), M10(48), M7(63), M9(61), M6(2)
Job19: M1(47), M3(23), M2(23), M5(82), M4(98), M8(66), M7(78), M9(100), M10(79), M6(32)
Job20: M2(13), M5(14), M1(90), M3(77), M4(80), M10(30), M8(31), M6(36), M7(51), M9(69)

TABLE 29. Job set data with process time used in example problems (large-scale for Job_Set18).

Job_Set18
Job1: M4(8), M5(73), M3(49), M6(24), M7(81), M2(68), M1(23), M13(69), M9(74), M11(45), M12(4), M15(59), M10(25), M8(70), M14(68)
Job2: M4(34), M3(33), M6(7), M2(69), M5(54), M7(18), M1(38), M9(28), M13(12), M15(50), M11(66), M8(81), M10(81), M14(91), M12(66)
Job3: M1(8), M7(20), M4(52), M5(83), M6(18), M3(82), M2(68), M8(50), M15(54), M12(6), M11(73), M14(48), M10(20), M9(93), M13(99)
Job4: M3(41), M1(72), M2(91), M5(52), M6(30), M4(1), M7(92), M14(52), M9(41), M10(45), M15(43), M13(97), M11(64), M12(71), M8(76)
Job5: M1(48), M2(44), M6(49), M7(92), M4(29), M3(29), M5(88), M15(14), M11(99), M9(22), M14(79), M10(93), M13(69), M12(63), M8(68)
Job6: M1(56), M7(42), M3(42), M4(93), M2(80), M5(54), M6(94), M13(80), M15(69), M12(39), M9(85), M11(95), M14(12), M10(28), M8(64)
Job7: M1(90), M5(75), M7(9), M2(46), M3(91), M4(93), M6(93), M15(77), M10(63), M12(50), M13(82), M14(74), M9(67), M8(72), M11(76)
Job8: M1(55), M3(90), M7(11), M4(60), M5(75), M2(23), M6(74), M12(54), M8(97), M13(32), M14(67), M11(15), M15(48), M9(100), M10(55)
Job9: M7(71), M6(64), M3(40), M1(32), M4(92), M2(59), M5(69), M14(68), M15(34), M13(71), M9(28), M10(94), M8(82), M11(1), M12(58)
Job10: M7(36), M5(46), M2(50), M6(87), M4(33), M3(94), M1(3), M15(60), M12(45), M14(84), M10(1), M9(38), M11(22), M13(39), M8(50)
Job11: M2(53), M1(34), M6(56), M7(97), M4(95), M5(32), M3(28), M15(48), M8(54), M13(98), M9(84), M10(77), M11(46), M14(65), M12(94)
Job12: M3(1), M6(97), M1(77), M5(82), M7(14), M2(18), M4(74), M15(52), M12(14), M13(93), M10(35), M9(34), M14(84), M11(6), M8(81)
Job13: M2(62), M1(86), M3(57), M7(80), M6(37), M4(94), M5(77), M8(72), M10(26), M12(41), M11(7), M9(56), M14(98), M15(67), M13(47)
Job14: M6(45), M4(30), M1(57), M7(68), M2(61), M3(34), M5(2), M8(57), M14(96), M10(10), M13(85), M15(42), M11(93), M9(89), M12(43)
Job15: M7(49), M5(53), M2(51), M3(4), M1(17), M6(21), M4(31), M11(45), M14(45), M10(63), M12(21), M9(4), M8(23), M15(90), M13(1)
Job16: M7(68), M6(18), M1(87), M4(6), M5(13), M3(9), M2(40), M9(83), M8(95), M13(27), M11(94), M15(68), M12(22), M14(28), M10(66)
Job17: M3(80), M7(14), M1(67), M6(15), M2(14), M4(97), M5(23), M9(45), M11(1), M12(5), M15(87), M8(34), M13(12), M10(98), M14(35)
Job18: M5(33), M3(20), M4(74), M7(20), M6(3), M1(90), M2(37), M14(56), M13(38), M9(7), M15(84), M10(100), M12(41), M11(6), M8(97)
Job19: M7(47), M5(63), M4(1), M1(28), M3(99), M2(41), M6(45), M15(60), M14(2), M8(25), M9(59), M10(39), M11(76), M12(89), M13(5)
Job20: M7(67), M3(46), M4(25), M2(2), M6(22), M5(8), M1(22), M14(64), M8(82), M13(99), M12(79), M11(87), M9(71), M10(24), M15(19)

past experience in the formulation of production plans into heuristic algorithms. At the same time, they can also inherit multi-dimensional experience in formulating proposals based on considerations of optimization objectives,

TABLE 30. Job set data with process time used in example problems (large-scale for Job_Set19).

Job_Set19
Job1: M6(8), M4(73), M1(69), M3(38), M7(6), M5(62), M2(78), M10(79), M9(59), M14(77), M12(22), M11(80), M13(58), M15(49), M8(48)
Job2: M4(34), M5(29), M3(69), M1(5), M6(63), M2(82), M7(94), M15(17), M12(94), M10(29), M11(5), M14(75), M8(15), M9(61), M13(61)
Job3: M2(52), M3(30), M1(25), M7(17), M4(46), M5(86), M6(3), M15(70), M12(34), M10(23), M11(68), M14(76), M9(53), M13(71), M8(9)
Job4: M3(50), M5(20), M4(24), M1(53), M2(97), M6(79), M7(92), M15(3), M13(52), M11(75), M10(74), M9(59), M8(75), M14(84), M12(99)
Job5: M3(15), M1(61), M4(47), M5(38), M7(49), M6(21), M2(6), M12(8), M9(71), M15(83), M14(24), M13(18), M10(33), M8(70), M11(100)
Job6: M5(48), M6(50), M3(66), M1(92), M7(2), M4(58), M2(23), M10(84), M9(66), M11(12), M8(36), M15(4), M13(88), M14(64), M12(12)
Job7: M4(29), M1(25), M7(44), M6(87), M3(42), M2(44), M5(86), M9(28), M11(86), M10(74), M15(77), M14(59), M13(94), M8(58), M12(16)
Job8: M5(31), M4(58), M1(94), M6(69), M3(44), M2(93), M7(92), M10(80), M9(63), M13(47), M14(3), M8(79), M12(39), M15(80), M11(75)
Job9: M2(69), M3(27), M1(76), M6(19), M7(86), M4(16), M5(31), M13(33), M10(69), M14(19), M11(43), M15(9), M12(37), M8(35), M9(24)
Job10: M3(75), M4(78), M7(41), M5(60), M6(59), M1(42), M2(60), M13(18), M9(31), M11(15), M8(54), M15(60), M10(20), M12(61), M14(69)
Job11: M5(89), M7(20), M2(27), M6(78), M4(2), M3(21), M1(55), M14(79), M12(77), M11(99), M10(70), M13(30), M8(97), M9(41), M15(98)
Job12: M7(1), M3(10), M5(84), M6(72), M1(14), M2(9), M4(51), M8(22), M15(65), M11(100), M14(65), M12(43), M9(10), M13(14), M10(19)
Job13: M6(50), M3(13), M4(49), M7(75), M2(42), M1(81), M5(89), M10(100), M15(54), M14(37), M11(7), M12(38), M9(25), M13(78), M8(79)
Job14: M3(44), M4(77), M6(26), M2(42), M5(9), M7(73), M1(60), M10(61), M11(85), M13(14), M12(92), M8(100), M15(49), M9(46), M14(12)
Job15: M3(72), M1(53), M2(43), M6(65), M7(59), M5(87), M4(13), M9(71), M13(25), M10(71), M11(89), M12(2), M8(76), M15(21), M14(12)
Job16: M3(60), M7(28), M6(33), M2(36), M1(6), M4(96), M5(48), M10(40), M12(79), M11(60), M9(39), M14(34), M8(54), M13(20), M15(52)
Job17: M6(82), M3(12), M4(11), M5(61), M2(21), M1(21), M7(34), M13(86), M15(53), M9(7), M10(4), M8(95), M11(62), M14(54), M12(82)
Job18: M6(72), M1(13), M4(46), M7(97), M2(87), M5(87), M3(11), M8(45), M15(85), M12(66), M9(43), M10(39), M14(34), M11(30), M13(55)
Job19: M2(39), M6(19), M1(19), M5(73), M7(63), M4(30), M3(69), M10(36), M8(13), M11(96), M13(27), M14(59), M15(76), M12(62), M9(14)
Job20: M2(7), M5(14), M4(79), M3(27), M7(43), M1(96), M6(24), M12(30), M8(27), M13(2), M9(69), M15(75), M14(34), M11(79), M10(96)

TABLE 31. Job set data with process time used in example problems (large-scale for Job_Set20).

Job_Set20
Job1: M4(8), M3(73), M2(79), M1(95), M7(69), M5(9), M6(5), M9(85), M10(52), M12(43), M15(32), M8(91), M11(24), M14(89), M13(38)
Job2: M7(45), M2(70), M5(84), M4(24), M6(18), M1(20), M3(71), M9(21), M8(60), M10(98), M11(70), M14(52), M13(34), M12(23), M15(52)
Job3: M7(16), M5(68), M2(85), M1(39), M6(40), M3(98), M4(61), M11(77), M8(60), M12(73), M10(66), M15(84), M9(16), M14(43), M13(88)
Job4: M1(72), M2(17), M4(68), M5(89), M3(94), M7(98), M6(56), M11(88), M14(27), M10(60), M13(61), M9(8), M8(88), M12(48), M15(65)
Job5: M7(78), M3(24), M6(28), M1(73), M5(21), M2(69), M4(52), M15(32), M9(83), M12(48), M11(29), M14(48), M13(92), M10(43), M8(82)
Job6: M5(54), M7(31), M6(14), M4(47), M1(82), M2(75), M3(4), M9(31), M13(72), M8(58), M10(45), M14(91), M15(31), M12(61), M11(27)
Job7: M5(88), M2(28), M6(92), M7(62), M4(93), M1(14), M3(65), M8(33), M10(44), M9(31), M15(32), M12(72), M14(47), M13(61), M11(34)
Job8: M1(52), M2(59), M6(98), M4(6), M3(19), M7(53), M5(39), M9(74), M13(48), M11(33), M14(49), M12(92), M8(22), M15(41), M10(37)
Job9: M1(2), M7(85), M4(34), M3(51), M5(97), M6(95), M2(73), M15(61), M10(28), M13(73), M9(21), M12(85), M8(75), M14(42), M11(7)
Job10: M6(94), M2(28), M1(77), M3(56), M7(79), M5(2), M4(82), M10(88), M11(93), M13(44), M15(5), M9(96), M8(34), M14(56), M12(41)
Job11: M3(15), M6(88), M7(18), M4(14), M2(82), M1(58), M5(33), M14(19), M11(42), M10(36), M15(57), M13(85), M8(3), M12(62), M9(36)
Job12: M4(30), M7(33), M1(13), M5(4), M3(74), M2(37), M6(78), M15(2), M14(56), M10(21), M11(61), M12(81), M8(18), M9(59), M13(62)
Job13: M6(40), M2(75), M7(45), M1(41), M4(97), M3(65), M5(92), M8(11), M13(44), M9(40), M10(100), M12(91), M15(66), M14(53), M11(27)
Job14: M2(83), M3(52), M1(84), M4(66), M6(3), M7(5), M5(71), M14(41), M11(42), M12(63), M13(50), M15(43), M9(3), M10(35), M8(18)
Job15: M5(44), M1(26), M2(59), M7(81), M3(84), M6(81), M4(91), M14(41), M8(42), M12(53), M9(63), M15(89), M10(15), M11(64), M13(40)
Job16: M2(46), M1(97), M6(67), M5(97), M4(71), M7(88), M3(69), M15(44), M13(20), M12(52), M14(34), M11(74), M9(79), M8(10), M10(87)
Job17: M4(71), M7(13), M5(100), M3(67), M2(57), M6(24), M1(36), M8(88), M15(79), M9(21), M10(86), M13(60), M12(28), M11(14), M14(3)
Job18: M1(97), M7(24), M3(41), M5(40), M2(51), M6(73), M4(19), M10(27), M13(70), M14(98), M11(11), M12(83), M8(76), M9(60), M15(12)
Job19: M6(88), M4(48), M2(33), M5(96), M7(10), M1(49), M3(52), M11(38), M14(49), M8(31), M13(94), M15(23), M10(7), M12(5), M9(4)
Job20: M3(85), M1(100), M6(51), M7(91), M2(21), M4(83), M5(30), M13(23), M10(48), M9(19), M12(47), M11(95), M8(23), M15(78), M14(22)

providing high-quality proposals in multiple dimensions. Finally, they can leverage the high computational power of computers to extract advantageous genes of decisions and improve the quality of proposals within a defined time frame.

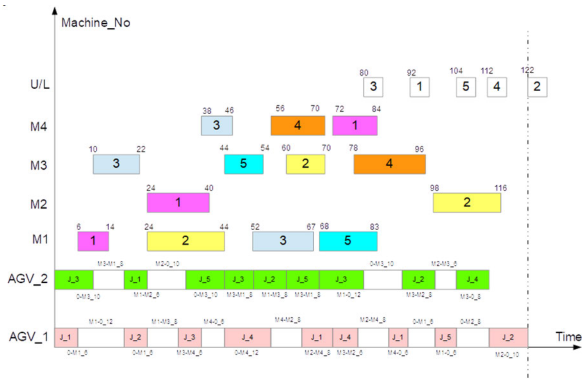


FIGURE 8. One of the pareto solutions in the case of Job_set1 under Layout_1.

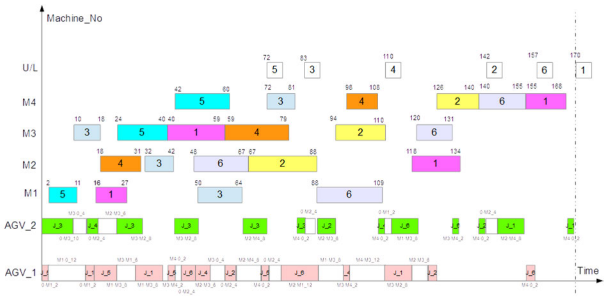


FIGURE 9. One of the pareto solutions in the case of Job_set10 under Layout_3.

In this research, the consumption of energy during transportation is only accounted for two states: loaded and empty, and there is no difference between two states. However, in actual scenarios, there are different sceneries for start-up, standby, load and empty transport [50]. And the energy consumption ratio during each scenery is different. Therefore, we can construct optimization models more closely corresponding to actual energy consumption states in the future, and further study the value of research results in production practice.

APPENDIX A
SMALL CASES DATA

(A) Layout information for four small-scale environments, the values in the table represent the transportation distance between the two nodes of the job-shop, and the transportation time between the nodes to be assisted by the transportation speed of AGV.

(B) Production information of 10 sets of jobs to be processed: Each job set consists of 5∼8 jobs to be processed, and from left to right the process of the job is indicated, while the brackets after the device indicate the process time.

APPENDIX B
LARGE CASES DATA

(A)Layout information for 2 large-scale environments: that Table. 20 representative the workshop layout data which contained 11 nodes that one node is the warehouse and other 10 nodes are the machines, and Table. 21 means the other workshop layout data which contained 16 nodes that 1 node is the warehouse and other nodes 15 are the machines.

(B) From Table. 22 to Table. 31 represent the production information of 10 sets of jobs to be processed in 2 large-scale environments: both are 20 jobs, but the processing information is different.

APPENDIX C
EXAMPLES OF OPTIMIZATION RESULTS

The Fig. 8 and 9 represent one of the pareto solutions in the case of Job_set1 under Layout-1 and Job_set10 under Layout-3.

REFERENCES

[1] L. Zhou, Z. Jiang, N. Geng, Y. Niu, F. Cui, K. Liu, and N. Qi, ''Production and operations management for intelligent manufacturing: A systematic literature review,'' *Int. J. Prod. Res.*, vol. 60, no. 2, pp. 808–846, Jan. 2022.

[2] 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.

[3] 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.

[4] W. Song, R. L. Storch, and Z. B. Zabinsky, ''An example for scheduling a chemical processing tank line,'' in *Proc. INRIA/IEEE Symp. Emerg. Technol. Factory Automat. (ETFA)*, Oct. 1995, pp. 475–482.

[5] I. Sabuncuoglu and D. L. Hommertzheim, ''Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system,'' *Int. J. Prod. Res.*, vol. 30, no. 5, pp. 1059–1079, May 1992.

[6] I. Sabuncuoglu and D. L. Hommertzheim, ''Experimental investigation of an FMS due-date scheduling problem: Evaluation of machine and AGV scheduling rules,'' *Int. J. Flexible Manuf. Syst.*, vol. 5, no. 4, pp. 301–323, Sep. 1993.

[7] I. Sabuncuoglu and D. L. Hommertzheim, ''Experimental investigation of an FMS due-date scheduling problem: An evaluation of due-date assignment rules,'' *Int. J. Comput. Integr. Manuf.*, vol. 8, no. 2, pp. 133–144, Mar. 1995.

[8] T. Sawik, ''A multilevel machine and vehicle scheduling in a flexible manufacturing system,'' *Math. Comput. Model.*, vol. 23, no. 7, pp. 45–57, Apr. 1996.

[9] M. Savelsbergh and M. Sol, ''Drive: Dynamic routing of independent vehicles,'' *Oper. Res.*, vol. 46, no. 4, pp. 474–490, Aug. 1998.

[10] S. Knust, ''Shop-scheduling problems with transportation,'' Univ. Osnabrück, Osnabrück, Germany, Tech. Rep., 1999.

[11] P. Brucker, S. Knust, and G. Wang, ''Complexity results for flow-shop problems with a single server,'' *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 398–407, Sep. 2005.

[12] P. Brucker, S. Knust, T. C. E. Cheng, and N. V. Shakhlevich, ''Complexity results for flow-shop and open-shop scheduling problems with transportation delays,'' *Ann. Oper. Res.*, vol. 129, nos. 1–4, pp. 81–106, Jul. 2004.

[13] J. Hurink and S. Knust, ''Makespan minimization for flow-shop problems with transportation times and a single robot,'' *Discrete Appl. Math.*, vol. 112, nos. 1–3, pp. 199–216, Sep. 2001.

[14] 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.

[15] A. Ham, ''Transfer-robot task scheduling in job shop,'' *Int. J. Prod. Res.*, vol. 59, no. 3, pp. 813–823, Feb. 2021.

[16] S. Huang, ''Optimization of job shop scheduling with material handling by automated guided vehicle,'' Iowa State Univ., Ames, IA, USA, Tech. Rep., 2018.

[17] T. J. Lin, C.-C. Chiu, E. Huang, and H.-M. Chen, ''A multi-fidelity model approach for simultaneous scheduling of machines and vehicles in flexible manufacturing systems,'' *Asia–Pacific J. Oper. Res.*, vol. 35, no. 1, pp. 1–20, 2018.

[18] P. Lacomme, M. Larabi, and N. Tchernev, "Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles," *Int. J. Prod. Econ.*, vol. 143, no. 1, pp. 24–34, May 2013.

[19] L. Deroussi, M. Gourgand, and N. Tchernev, "A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles," *Int. J. Prod. Res.*, vol. 46, no. 8, pp. 2143–2164, Apr. 2008.

[20] 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.

[21] T. F. Abdelmaguid, A. O. Nassef, B. A. Kamal, and M. F. Hassan, "A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles," *Int. J. Prod. Res.*, vol. 42, no. 2, pp. 267–281, Jan. 2004.

[22] A. Tabatabaei, F. Torabi, and T. Paitoon, "Simultaneous scheduling of machines and automated guided vehicles utilizing heuristic search algorithm," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Las Vegas, NV, USA, Jan. 2018, pp. 54–59.

[23] S. Karimi, Z. Ardalan, B. Naderi, and M. Mohammadi, "Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm," *Appl. Math. Model.*, vol. 41, pp. 667–682, Jan. 2017.

[24] A. Ham, "Transfer-robot task scheduling in flexible job shop," *J. Intell. Manuf.*, vol. 31, no. 7, pp. 1783–1793, Oct. 2020.

[25] D. B. M. M. Fontes and S. M. Homayouni, "Joint production and transportation scheduling in flexible manufacturing systems," *J. Global Optim.*, vol. 74, no. 4, pp. 879–908, Aug. 2019.

[26] C. Kui, B. Li, and W. Wenya, "Research on integrated scheduling of AGV and machine in flexible job shop," *J. Syst. Simul.*, vol. 34, no. 3, pp. 461–469, 2022.

[27] W. Yue, S. Yuchuan, L. Xiangfei, and L. Qi, "An improved differential evolution algorithm for simultaneous scheduling of machines and AGVs in an FMS," *J. Chongqing Univ.*, vol. 44, no. 12, pp. 116–129, 2021.

[28] Z. Wang and Y. Wu, "An ant colony optimization-simulated annealing algorithm for solving a multiload AGVs workshop scheduling problem with limited buffer capacity," *Processes*, vol. 11, no. 3, p. 861, Mar. 2023, doi: 10.3390/pr11030861.

[29] H. Nabovati, H. Haleh, and B. Vahdani, "Multi-objective invasive weeds optimization algorithm for solving simultaneous scheduling of machines and multi-mode automated guided vehicles," *Eur. J. Ind. Eng.*, vol. 14, no. 2, pp. 165–188, 2020.

[30] M. R. Mahalakshmi, "Optimization algorithm for minimizing the earliness or tardiness of automated guided vehicles using artificial immune system," *Int. J. Appl. Eng. Res.*, vol. 12, no. 22, pp. 11913–11919, 2017.

[31] C. Lei, N. Zhao, S. Ye, and X. Wu, "Memetic algorithm for solving flexible flow-shop scheduling problems with dynamic transport waiting times," *Comput. Ind. Eng.*, vol. 139, Jan. 2020, Art. no. 105984.

[32] G. Li, B. Zeng, W. Liao, X. Li, and L. Gao, "A new AGV scheduling algorithm based on harmony search for material transfer in a real-world manufacturing system," *Adv. Mech. Eng.*, vol. 10, no. 3, pp. 1–13, 2018.

[33] X. Feng and Z. Xu, "Integrated production and transportation scheduling on parallel batch-processing machines," *IEEE Access*, vol. 7, pp. 148393–148400, 2019.

[34] M. Mousavi, H. J. Yap, and S. N. Musa, "A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS," *Int. J. Simul. Model.*, vol. 16, no. 1, pp. 58–71, Mar. 2017.

[35] M. Dai, D. Tang, A. Giret, and M. A. Salido, "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints," *Robot. Comput.-Integr. Manuf.*, vol. 59, pp. 143–157, Oct. 2019.

[36] W.-Y. Ku and J. C. Beck, "Mixed integer programming models for job shop scheduling: A computational analysis," *Comput. Oper. Res.*, vol. 73, pp. 165–173, Sep. 2016.

[37] F. Neri et al., *Handbook of Memetic Algorithms*, vol. 379. Heidelberg, Germany: Springer, 2012, pp. 9–27.

[38] Z. Chaoyong, D. Xing, W. Xiaojuan, L. Xinyu, and L. Qiong, "Improved NSGA-II for the multi-objective flexible job-shop scheduling problem," *J. Mech. Eng.*, vol. 46, no. 11, pp. 156–164, 2010.

[39] M. Nawaz, E. E. Enscore Jr., and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, Jan. 1983.

[40] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *Comput. Technol. J.*, vol. 52, no. 2, pp. 26:1–26:38, 2019.

[41] Y. Yuan and H. Xu, "Multiobjective flexible job shop scheduling using memetic algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 336–353, Jan. 2015.

[42] Y. Tian, "Evolutionary large-scale multi-objective optimization: A survey," *ACM Comput. Surv.*, vol. 54, no. 8, p. 174, 2021.

[43] R. Ruiz, C. Maroto, and J. Alcaraz, "Two new robust genetic algorithms for the flowshop scheduling problem," *Omega*, vol. 34, no. 5, pp. 461–476, Oct. 2006.

[44] D. C. Montgomery, *Design and Analysis of Experiments*. Hoboken, NJ, USA: Wiley, 2017.

[45] E. Teymourian, V. Kayvanfar, G. M. Komaki, and M. Khodarahmi, "An enhanced intelligent water drops algorithm for scheduling of an agile manufacturing system," *Int. J. Inf. Technol. Decis. Making*, vol. 15, no. 2, pp. 239–266, Mar. 2016.

[46] V. Kayvanfar, M. Zandieh, and M. Arashpour, "Hybrid bi-objective economic lot scheduling problem with feasible production plan equipped with an efficient adjunct search technique," *Int. J. Syst. Sci., Oper. Logistics*, pp. 1–24, Apr. 2022.

[47] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[48] D. Lei, "A Pareto archive particle swarm optimization for multi-objective job shop scheduling," *Comput. Ind. Eng.*, vol. 54, no. 4, pp. 960–971, May 2008.

[49] C. Shen and Y. L. Chen, "Blocking flow shop scheduling based on hybrid ant colony optimization," *Int. J. Simul. Model.*, vol. 19, no. 2, pp. 313–322, Jun. 2020.

[50] V. K. Chaela, A. K. Chanda, and A. Surjit, "Automatic guided vehicle systems in the flexible manufacturing system—A review," *Int. J. Ind. Eng.*, vol. 26, no. 5, pp. 737–765, 2019.

CHENGFENG PENG received the B.S. degree in industrial engineering from the Jiangxi University of Technology, in 2014, and the Ph.D. degree in mechanical engineering from the Guangdong University of Technology, in 2021. He is currently an Assistant Professor with the School of Physics, Electronics and Electrical Engineering, Xiangnan University. His research is devoted into the theory and developments of production planning and control, evolutional algorithms, multi-objective optimization in complex production systems, intelligent manufacturing, and enterprise information management.

ZHANTAO LI received the B.S. and Ph.D. degrees in mechanical engineering from the Guangdong University of Technology, in 2006 and 2012, respectively. He is currently an Associate Professor with the Department of Electromechanical Engineering, Guangdong University of Technology. His research is devoted into the theory and developments of production planning and control, evolutional algorithms, multi-objective optimization in complex production systems, and enterprise information management systems.

HONGYANG ZHONG received the B.S. and Ph.D. degrees in mechanical engineering from the Guangdong University of Technology, in 2014 and 2022, respectively. He is currently an Assistant Professor with the Electronic Information School, Foshan Polytechnic. His research interests include production planning and control and discrete event simulation.

XIANG LI was born in Chenzhou, Hunan, China, in 1982. He received the Ph.D. degree from the Guangdong University of Technology, China. He is currently an Associate Professor with the School of Physics, Electronics and Electrical Engineering, Xiangnan University, where he is also the Deputy Director of the Academic Affairs Office. His research interests include computer integrated manufacturing systems, system modeling, analysis and optimization, and control system simulation.

ANPING LIN received the B.S. and M.S. degrees in marine engineering from Dalian Maritime University, Dalian, China, in 2004 and 2007, respectively, and the Ph.D. degree in control science and engineering from Hunan University, Changsha, in 2019.

From 2007 to 2014, he was a Teacher with Guangdong Ocean University, Zhanjiang, China. He is currently an Assistant Professor with the School of Physics, Electronics and Electrical Engineering, Xiangnan University, Chenzhou, China. He has authored a few refereed articles, including *Information Sciences*, *Swarm and Evolutionary Computation*, and *Applied Soft Computing*. His main research interest includes evolutionary computation and application.

YONG LIAO was born in Loudi, Hunan, China, in 1990. He received the Ph.D. degree from the Guangdong University of Technology, China. He is currently an Assistant Professor with the School of Physics, Electronics and Electrical Engineering, Xiangnan University. His research interest includes intelligent manufacturing system design and optimization.