An Integrated Production and Transportation Scheduling Method in Hybrid Flow Shop

Wangming Li
Huazhong University of Science and Technology - Main Campus: Huazhong University of Science and Technology

Dong Han
Huazhong University of Science and Technology - Main Campus: Huazhong University of Science and Technology

Liang Gao
Huazhong University of Science and Technology - Main Campus: Huazhong University of Science and Technology

Xinyu Li (lixinyu@mail.hust.edu.cn)
Huazhong University of Science and Technology

Yang Li
Huazhong University of Science and Technology - Main Campus: Huazhong University of Science and Technology

Original Article

Keywords: Hybrid flow shop, Integrated scheduling, Task pool, Hybrid algorithm

Posted Date: March 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-265121/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Version of Record: A version of this preprint was published at Chinese Journal of Mechanical Engineering on February 18th, 2022. See the published version at https://doi.org/10.1186/s10033-022-00683-7.
Title page

An integrated production and transportation scheduling method in hybrid flow shop

Wangming Li, born in 1997, is currently a master candidate at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his bachelor degree from Huazhong University of Science and Technology, China, in 2019. His main research interests are scheduling problem and algorithm optimization.

Dong Han, born in 1997, is currently a master candidate at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. She received his bachelor degree from Huazhong University of Science and Technology, China, in 2019. Her main research interests are deep learning and algorithm optimization.

Liang Gao, born in 1974, is currently a professor at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his PhD degree in mechatronic engineering from Huazhong University of Science and Technology, China, in 2002. His main research interests are intelligent optimization method and its application in design and manufacturing.

Xinyu Li, born in 1985, is currently a professor at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his PhD degree in industrial engineering from Huazhong University of Science and Technology, China, in 2009. His main research interests are intelligent manufacturing systems, shop scheduling, intelligent optimization and machine learning.

Yang Li, born in 1996, is currently a doctor at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his bachelor degree from Huazhong University of Science and Technology, China, in 2018. His main research interests are flow shop scheduling and algorithm optimization.

Corresponding author: Xinyu Li   E-mail: lixinyu@mail.hust.edu.cn
An integrated production and transportation scheduling method in hybrid flow shop

Wangming Li¹ • Dong Han¹ • Liang Gao¹ • Xinyu Li¹ • Yang Li¹

Received June xx, 201x; revised February xx, 201x; accepted March xx, 201x
© Chinese Mechanical Engineering Society and Springer-Verlag Berlin Heidelberg 2017

Abstract: The connection between production scheduling and transportation scheduling is getting closer in smart manufacturing system, both of which are summarized as NP-hard problems. However, only a few studies have considered them simultaneously. This paper studies the integrated production and transportation scheduling problem (IPTSP) in hybrid flow shop, which is an extension of the hybrid flow shop scheduling problem (HFSP). In this problem, the transfer tasks of jobs are performed by a certain number of automated guided vehicles (AGV). In addition to the production scheduling on machines, we consider the transportation scheduling on AGVs as the part of the optimization process. To solve it, we make some preparation (including the establishment of task pool, the new solution representation and the new solution evaluation), which can help algorithm efficiently find satisfactory solutions while appropriately limiting the search space. Then, an effective genetic tabu search algorithm is used to minimize the makespan. Finally, two groups of instances are designed and three types of experiments are conducted to evaluate the performance of proposed method. The results show that the proposed method can achieve good results, showing the effectiveness of the presented approach.

Keywords: Hybrid flow shop • Integrated scheduling • Task pool • Hybrid algorithm

1 Introduction

Due to the development of industry, scheduling plays a crucial role in modern manufacturing systems. As a branch of flow shop, the hybrid flow shop is widespread in modern industries, including electronics [1], textile [2], steelmaking [3] and petrochemical industries [4].

In mechanical manufacturing, production scheduling and transportation scheduling are two vital parts [5]. Optimizing both of them is a core task of advanced manufacturing and modern management, which not only allocates tasks but also affects the utilization level of resources and energy [6].

The importance of production scheduling problem and transportation scheduling problem has been emphasized by many researchers [7-11]. For the literature of production scheduling problem in hybrid flow shop (also called hybrid flow shop scheduling problem, HFSP), most of the researchers didn’t consider the transport procedure between machines or put it as fixed value into the setup time. However, more and more flexible transporters like automated guided vehicle (AGV) are used to perform the transfer tasks in the modern factory, which obviously improves the productivity of manufacturing enterprises [9].

On the other hand, the use of AGVs also brings uncertainty and complexity to the current scheduling scheme. For example, unprocessed jobs can be processed on a set of alternative machines at a specific stage. All of these can cause uncertainty in transfer time during scheduling. As a result, in many manufacturing industries that are sensitive to transport time and limited transport resources, they attaches more and more importance to considering production scheduling and transportation scheduling integratedly.

So, this paper integrates the production and transportation scheduling in hybrid flow shop with
An integrated production and transportation scheduling method in hybrid flow shop

identical machines, which is an extension of the hybrid flow shop scheduling problem (HFSP). Transfer tasks between machine and machine or machine and warehouse are performed by a certain number of automated guided vehicles (AGV). And the goal of the integrated scheduling problem (IPTSP) is finding optimal processing sequence of jobs on machines and optimal transport sequence of jobs on AGVs simultaneously.

Production scheduling and transportation scheduling are both well-known NP-hard problems [17-18]. There are several researches that studied the coordination between production scheduling and transportation scheduling. Bilge and Ulusoy [12] assigned the transportation task to AGVs while scheduling the processing sequence of jobs on the machine. Amir and Pedram [13] addressed a permutation flow-shop scheduling problem with a finite number of transporters carrying jobs from each machine to its subsequent machine. Nishi et al. [14] used a bilevel decomposition algorithm to solve the simultaneous scheduling and conflict-free routing problems for AGVs. Elmi et al. [15] addressed the robotic scheduling problem considering multiple part types, unrelated parallel machines, multiple robots in blocking hybrid flow shop. Zabihzadeh et al. [16] used ant colony optimization (ACO) algorithm and genetic algorithm (GA) to solve flexible flow shop scheduling problem with robotic transportation and release time.

However, in all related surveys mentioned above, they considered all stages of each job and put them as a long sequence to be optimized. When the long sequence was worked as the code in their algorithms, it might make the algorithms search too much solution space and hardly get a satisfactory solution within reasonable computation time. Therefore, this paper proposes a new method for integrated production and transportation scheduling problem in hybrid flow shop environment, including the establishment of task pool, the new solution representation, the new solution evaluation and so on. Based on it, two scheduling problems mentioned above can be treated together. Then, a genetic algorithm with tabu search is applied to solve the integrated scheduling problem.

This paper is organized as follow. Section 2 presents the notation and description of problem. Section 3 introduces some preparation for solving the IPTSP. Section 4 describes the details of the hybrid algorithm. Section 5 shows the experimental design and results. Finally, Section 6 gives the conclusion and future work.

2 Problem description and formulation

In hybrid flow shop environment, a set of $n$ jobs needs to be processed at $S$ stages. Each stage $j$ has $M_j$ identical parallel machines, while $M_j \geq 2$ for at least one stage. Once a job has completed processing at a certain stage, it needs to be transferred to the machine of its next stage by AGV. $R$ identical AGVs are responsible for these transferring tasks. The object of scheduling is to determine the assignment of machines and AGVs at each stage for each job, the sequence of jobs on machines and the sequence of transferring tasks on AGVs, such that the makespan is minimized.

A small example is shown. There are 3 jobs to be processed without preemption on 4 machines. Each job needs to go through 3 processing stages. The number of identical machines in each stage is $\{1, 2, 1\}$. The number of AGVs is 2. The processing time of each job at each stage is shown in Table 1. The transport time between machine and machine (or warehouse) is shown in Table 2. Figure 1 shows a Gantt chart of a scheduling scheme with makespan 40.

Figure 1 An example for the integrated scheduling problem

| Time(job/stage) | Stage 1 | Stage 2 | Stage 3 |
|----------------|---------|---------|---------|
| Job1           | 6       | 8       | 10      |
| Job2           | 5       | 5       | 4       |
| Job3           | 6       | 6       | 8       |

Table 1 Processing time of 3 jobs

| Time (from/to) | Warehouse | Machine 1 | Machine 2 | Machine 3 | Machine 4 |
|----------------|-----------|-----------|-----------|-----------|-----------|
| Warehouse      | 0         | 2         | 4         | 6         | 6         |
| Machine 1     | 2         | 0         | 2         | 4         | 4         |
| Machine 2     | 4         | 2         | 0         | 2         | 2         |
| Machine 3     | 6         | 4         | 2         | 0         | 4         |
| Machine 4     | 6         | 4         | 2         | 4         | 0         |

Table 2 Transport time between locations
3 Some preparation for solving IPTSP

Under the same scale of jobs and machines, it’s obvious that the solution space of integrated scheduling problem is much larger than that of production scheduling or transportation scheduling. For better algorithm design in next section, some preparation are considered to be completed. Firstly, the task pool is introduced to transform scheduling into task selection and assignment, which can simplify the IPTSP. Secondly, a new way of solution representation according to the proposed task pool is proposed, which provides an encoding method for the IPTSP that can be operated by subsequent algorithms. Finally, the solution evaluation can evaluate the makespan of each code. By this way, all of them can help algorithm efficiently find satisfactory solutions while appropriately limiting the search space.

3.1 Establishment of task pool

In the IPTSP, each job at each stage requires a transportation task which takes it from previous machine (or warehouse) to current machine. And the number of existing transportation tasks does not exceed the total number of jobs at the same time. According to these characteristics, this paper establishes a specific set of transportation tasks, called task pool. Then, the integrated scheduling problem can be solved through the procedure that each time the AGVs execute the tasks in the task pool in a specific order until the task pool is emptied.

Specifically, a task can be described as:

$$\text{Task} = (\text{job\_no}, \text{stage\_no}, s, \text{from\_location}) \tag{1}$$

Where job\_no is the index number of the job involved, stage\_no is the current stage number of the job involved, s is the earliest start time of the task, from\_location is the starting location of the task (if stage\_no = 1, it represents the warehouse; else, it represents the processing machine of job at stage stage\_no - 1).

Here, task pool is a collection of all tasks to be scheduled. Initially, in task pool, the number of tasks is equal to the number of jobs, the job\_no of all tasks represent all jobs to be scheduled, the stage\_no of all tasks represent stage one, the s of all tasks are zero, the from\_location of all tasks represent warehouse. Once a task in task pool is completed by AGV, if the stage\_no of the task is not equal to the last stage of the job, the completed task (Taskold) will be removed from the task pool. At the same time, a new task (Tasknew) representing the next stage of the same job is added to the task pool. The old and new tasks satisfy the following relationship:

- The job\_no\_new is equal to job\_no\_old.
- The stage\_no\_new represents the next stage of stage\_no\_old.
- The s\_new is equal to the completion time of the job i at stage j involved in the Task\_old.
- The from\_location\_new is the processing machine of the job i at stage j involved in the Task\_old.

Where, job\_no\_new and stage\_no\_new is the job\_no and stage\_no of Task\_new, job\_no\_old and stage\_no\_old is the job\_no and stage\_no of Task\_old, s\_new and from\_location\_new is the s and from\_location of Task\_new. A simple example is shown in Figure 2. Once the old task Task\_old is completed and stage one is not the last stage of job2, a new Task\_new which represent the next stage of the job is added in task pool. In the new task, "6" represents the completion time of job2 on machine2 at stage one (also the earliest start time of the new task). And "machine2" represents the processing machine selected for job2 at stage one (also the starting location of the new task).

After this, the object of scheduling is to assign the tasks in the task pool to each AGV in a certain order and then to schedule the trip and processing, making the makespan smaller. While the task pool is emptied, the scheduling is complete.

![Figure 2](image_url) Simple description of the task pool

3.2 Solution representation

A new way of solution representation according to the proposed task pool is proposed, which provides an encoding method for the IPTSP that can be searched efficiently by subsequent algorithms.

Under some task selection rules based on time target (like: FCFS, EDT, GWTQ and so on [19]), we find that the assignment of AGVs and the order of tasks transporting at the first stage have a great impact on the makespan. This is because of the fact that the earliest start time of tasks in task pool are all zero and starting locations are all warehouses (at the first stage). Therefore, we consider transforming the assignment of AGVs and the order of tasks transporting at the first stage into a coding sequence to be optimized. For that a new solution representation is introduced, based on tasks in task pool at the first stage. The new solution representation is composed of two parts:

- Transport sequence vector (also called v1)
Transporter assignment vector (also called $v_2$)

Transport sequence vector $v_1$ represents the order of transport for each task at the first stage. Figure 3 illustrates a transport sequence vector. For example, the transport sequence shown in Figure 3 can be translated into a list of ordered tasks below: Task3 > Task2 > Task1 > Task4 > Task5.

| Position | Priority $u$ | Task Indicated | Transport sequence $v_1(u)$ |
|----------|--------------|----------------|-----------------------------|
| 1        | 1            | Task1 Task2 Task3 Task4 Task5 |
| 2        | 2            | 3 2 1 4 5 |

Figure 3 Illustration of the transport sequence vector

Each transporter assignment vector $v_2$, $v_2(u)$ represents the AGV selected for the $Task_m$ indicated at position $u$. Figure 4 illustrates a transporter assignment vector. For example, position 3 indicates $Task_3$, and $v_2(3)$ represents the AGV assigned for $Task_3$.

| Position | Task Indicated | Transporter Assignment $v_2(u)$ |
|----------|---------------|--------------------------------|
| 1        | Task1 Task2 Task3 Task4 Task5 |
| 2        | 2 2 1 2 1 |

Figure 4 Illustration of the transporter assignment vector

With the new solution representation, the assignment of AGVs and the sequence of tasks at the first stage have been confirmed. Furthermore, such algorithm applying requires a makespan evaluation with a computationally fast sub procedure. It will be presented in the next section.

3.3 Solution evaluation

Using the proposed solution representation method, the makespan evaluation is described as follows:

- At stage one, the transport sequence and the assignment of AGVs are determined according to the two-vector representation;
- At stage $j$ ($j > 1$), a task is assigned to the earliest idle AGV with a specific task selection rule. Repeat the above process until the task pool is emptied.

Among them, after the task is selected and assigned to an AGV, the processing machine (also the destination of the task) at this stage is selected according to the improved First Available Machine (FAM [20]) rule. When the task is completed, if the stage involved in this task doesn’t represent the last stage, the task is removed from the task pool and a new task is added (refer to section 3.1 for the details).

The steps are shown below:

Step 1. Schedule the tasks in task pool at stage one:

1a). Read the position information in $v_1$ from left to right, and get the corresponding task $Task_m$ to be scheduled;
1b). Get other information about $Task_m$. For example, we can know the assigned AGV $R_v$ for the task in $v_2$, the involved job $J_i$, the processing time $p_{i,v}$, the earliest start time of the task $st$, the starting location $M_i$;
1c). Plan the empty trip (from the destination $M_v$ of the previous task to machine $M_v$) for $R_v$. The start time of the empty trip is earliest idle time of $R_v$. The arrival time of the empty trip is calculated as follow:

$$CT_{i,v} = AIT_v + p_{i,v},$$

(2)

Where, $p_{i,v}$ represents the transport time from location $M_v$ to location $M_v$, $AIT_v$ is the earliest idle time of $R_v$;
1d). Select the processing machine for $J_i$ at stage one according the improved FAM rule. The estimated completion time $C_{i,v}'$ of $J_i$ at stage one on each available machine $M_p$ is calculated as follow:

$$C_{i,v}' = \max(\max(CT_{i,v}, st) + p_{i,v} + MIT_p, p_i) + p_i,$$

(3)

Where, $MIT_p$ represents the earliest idle time of machine $M_p$, $p_{i,v}$ represents the transport time from location $M_v$ to location $M_p$, $p_i$ represents the processing time of $J_i$ at stage one.

Then, select the machine $M_q$ with the smallest estimated completion time as the processing machine for $J_i$ at stage one.
1e). Plan the loaded trip (from location $M_k$ to location $M_q$) for $R_v$. The start time $ST_{i,v}$ of the loaded trip is the maximum between the earliest start time of the task and the arrival time of the empty trip:

$$ST_{i,v} = \max(CT_{i,v}, st),$$

(4)

And the arrival time of the loaded trip is calculated as follow:

$$CT_{i,v} = ST_{i,v} + p_{i,v},$$

(5)

Where, $p_{i,v}$ represents the transport time from location $M_k$ to location $M_q$;
1f). Plan $J_i$ to be processed on $M_q$. The start time $Si$ of processing on machine $M_q$ is the maximum between the earliest idle time of machine $M_q$ and the arrival time of the loaded trip:
\[ S_{il} = \max(C_{il}, M_{il}), \quad (6) \]

Where, \( M_{il} \) represents the earliest idle time of machine \( M_q \);

1g). Update the earliest idle time of machine \( M_q \) and AGV \( R_v \) as follow:

\[ M_{il} = S_{il} + p_{il}, \quad (7) \]

\[ A_{il} = C_{il}, \quad (8) \]

1h). Update the task pool referred to section 3.1;

1i). Repeat steps (1a)–(1h) until the last position of two-vector representation is read.

Step 2. Schedule the tasks in task pool at stage \( j (j > 1) \):

2a). Select the AGV \( R_u \) with the earliest idle time (\( A_{IT} \));

2b). Select a task from task pool as the next task for \( R_u \);

According to first come first served (FCFS) rule, the way that selecting the task \( Task_{u,m} \) with the smallest \( st \) is applied;

2c). Plan the empty trip and loaded trip for \( R_u \), select processing machine, plan the job processing and update information. All of these procedures are similar with steps from (1b) to (1h);

2d). Repeat steps (2a)–(2c) until the task pool is emptied.

Step 3. Get the makespan of current two-vector solution representation.

3.4 Advantages of the proposed solution representation

The proposed solution representation has major differences with that of the literatures. Literatures [15-16] represent a solution by three long vectors which considered the sequence of operations, machine assignment and transporter assignment for each operation at the all stages from a global view. They had proved the effectiveness of it and succeed in many cases. However, when the size of problem becomes larger, it may cause the algorithm to perform a lot of invalid searches and hardly to find an optimal solution.

The proposed solution representation uses two vectors representing the order sequence and assignment of AGVs at the first stage which has a big impact on result. The heuristic rules select tasks for corresponding AGV and select machine for jobs at the other stage. The advantage of it is that each two-vector representation can be transformed into feasible scheduling scheme. And when applied to algorithms, it can limit the search space within a considerably range. For large-scale problems, a satisfactory solution can be obtained easily within a limited time. Detailed comparative experiments will be presented in Section 5.

4 Proposed Genetic algorithm with tabu search for IPTSP

Genetic algorithm (GA) is a well-known meta-heuristic algorithm proposed by Holland inspired by the laws of biological evolution in nature. Tabu search (TS) is a local search algorithm proposed by Glover [21] to simulate human memory function. In this paper, the tabu search algorithm is nested into GA for improving offspring individuals in each generation. The framework of the method is shown in Figure 5.
4.1 Population initialization
The chromosomes in GA are corresponding to the solutions or gantt charts of the integrated scheduling problem. The method of chromosome representation and decoding is as same as that described in section 3. In order to ensure the diversity of the population, the algorithm initialize the individuals in the initial population randomly.

4.2 Crossover operator
In GA, the crossover operator determines the way that parents produce new individual, and promotes the algorithm’s global search capabilities. In this paper, two crossover operators have been adopted for the two vectors.

The first one is the position-based crossover (PBX) for transport sequence vector ($v_1$). The basic procedure of PBX is described as follow (two parents are noted as P1 and P2; two offspring are noted as C1 and C2):

Step1. Randomly generate several gene positions, and C1 and C2 respectively inherit the genes of the corresponding gene positions from P1 and P2;

Step2. C1 and C2 inherit the remaining genes from another parent (P2 and P1) on the unselected gene position.

The second one is the multi-point crossover (MPX) for transporter assignment vector ($v_2$). The basic procedure of MPX is described as follow (two parents are noted as P1 and P2; two offspring are noted as C1 and C2):

Step1. Randomly generate a sequence composed of 0 and 1, and the length of the sequence is equal to the length of $v_2$;

Step2. Select the same genes in P2 and P1 corresponding to position 1 in the sequence, and copy them to C1 and C2, that is, exchange the assigned transporter;

Step3. Keep the remaining genes in P1 and P2 to C1 and C2, thus generating offspring C1 and C2.

4.3 Mutation operator
In GA, the mutation operator is used to make perturbations on chromosomes in order to maintain the algorithm’s local search capabilities. In this paper, two mutation operators have been adopted for the two vectors.

The first mutation operator is used for transport sequence vector ($v_1$), which selects two genes randomly and inserts the back one before the front one or the front one after the back one.

The second mutation operator is used for transporter assignment vector ($v_2$), which selects one gene randomly and change the value of this selected gene to the other AGV.

4.4 Neighborhood structure
In TS, neighborhood structure is a mechanism for generating new solutions by making small disturbances to the current solution. In this paper, four ways of
neighborhood structure are adopted.

1. Binary exchange: select two points randomly in chromosome and reverse the order of all genes between these two points.
2. Two points exchange: select two points randomly in chromosome and exchange the gene value of these two points.
3. One point insert: select two point randomly and insert the back one before the front one.
4. Transporter change: select one point randomly and change the AGV assignment of this gene to the other AGV.

### 4.5 Tabu list

The purpose of the tabu list is to avoid roundabout searches and guide the algorithm to better explore other areas of the solution space. The length of the tabu list is the tenure of the subject staying in the tabu list. If the length is \( L \), the tabu list can be expressed as a ring table composed of \( L \) subjects. Whenever a new subject is added to the tabu list, it is possible to overwrite one of the oldest elements with this new subject.

### 4.6 Termination criterion

The termination criterion determines whether the algorithm should stop. In this paper, the GA with TS terminates when the number of iterations reaches to the maximum iterations (\( \text{MaxIter} \)); TS terminates when the number of iterations reaches to the maximum iterations (\( \text{MaxTSIter} \)).

### 5 Experiments and computational results

Since there is no corresponding benchmark for the integrated scheduling problem in hybrid flow shop environment, we design two groups of instances for the problem in this section. Then we adapt three types of experiments to verify the proposed method. Finally, we make some analysis based on the results. All of the experiments have been coded in C++ and run on Intel Core i5 2.3 GHz PC with 8 GB memory.

#### 5.1 Instances design

This paper has designed two groups of instances. And all of them can be found at the website [29].

#### 5.1.1 Group 1

Literature [22] specializes in the benchmark of HFSP. We take some instances in the literature [22] as a part of input information of the instances of the integrated scheduling problem, and the corresponding transportation time between machines is generated according to number of machines. The instances of proposed benchmark is divided into small-size and large-size according to the number of jobs, as follows:

- **Small-size**: number of jobs: \{10, 20, 30\}; number of stages: \{5, 10\}; number of AGVs: \{2, 4, 6\};
- **Large-size**: number of jobs: \{80, 160\}; number of stages: \{5\}; number of AGVs: \{4, 6, 8\}.

When generating the problems, an important characteristic considered is the relative magnitude of the travel times and the processing times [23]. We consider using a variable \( \alpha \) to distinguish different types of instances, and further generate large-\( \alpha \) type and small-\( \alpha \) type in large/small-size instances. The \( \alpha \) represents the ratio between the average machine-to-machine transportation time and the average processing time of the instance. The value of \( \alpha \) in each instance is calculated as follows:

\[
\alpha = \frac{(\sum \sum p_{ij})/(n*s)}{(\sum \sum pt_{pq})/(m+1)^2},
\]

Where, \( p_{ij} \) is the processing time of job \( i \) at stage \( j \); \( pt_{pq} \) is the transportation time of location \( p \) to location \( q \); \( n \) is the number of jobs; \( s \) is the number of stages; \( m \) is the number of machines.

#### 5.1.1 Group 2

Zabihzadeh and Rezaeian [16] have researched similar problems to this paper. They gave the parameters which were used to generate the instances, but did not give specific instances they used. In this group, parameters are generated randomly based on Zabihzadeh and Rezaeian’s research [16]. For each job, standard processing time, unloading, transferring and loading time at each stage are generated from the uniform distribution \( U[10, 100], U[5, 20], U[5, 20], U[5, 20] \), respectively. The match between the number of jobs, the number of stages, and the number of AGVs still refer to their research.

#### 5.2 Experimental Setup

#### 5.2.1 Experiment 1: Comparison of solution representations

Using the same genetic algorithm in literature [16], under the same number of iterations and other parameters, this paper solves proposed two groups of instances in three different solution representations (coding and decoding methods) in order to verify the effectiveness of the proposed one:
5. Mode1: Use the coding and decoding method of the HFSP problem in literature [27], and whenever the job is scheduled to be processed on the machine, use the AGV assignment rule in literature [28] to allocate the appropriate AGV to complete the corresponding transportation task.

- Mode2: Use the same coding and decoding method of the integrated scheduling problem with literature [16]. Blocking and release time have been considered in literature [16], both of which are ignored here.

- Mode3: Use the coding and decoding method proposed in Section 3.

5.2.2 Experiment 2: Comparison of selection rules

According to Section 3.3, the task selection rule has an important impact on solution evaluation. Here, four common and classic rules are used for experiment and comparison, each of which is worked as a part of solution evaluation in the proposed GATS independently [19].

- Rule1: First Come First Served (FCFS). Each time, select the task with the minimum earliest start time (st);
- Rule2: Longest Time Between Arrival (LTBA). Each time, select the task with the minimum difference between the AGV’s estimated arrival time and the earliest start time;
- Rule3: Shortest Travel Distance Rule (STD). Each time, select the task closest to the current location of the AGV;
- Rule4: Greatest Waiting Time in Queue (GWTQ). Each time, select the task with the longest waiting time in the buffer.

5.2.3 Experiment 3: Comparison of algorithms

In order to verify the effectiveness of the hybrid algorithm, the following five algorithms have been selected and programmed to solve the two groups of instances and compared with the hybrid GATS algorithm.

- Genetic algorithm (GA);
- Simulated annealing algorithm (SA);
- Artificial bee colony algorithm (ABC) [24];
- Grey wolf optimizer algorithm (GWO) [25];
- Migrating birds optimization algorithm (MBO) [26].

5.3 The experimental results

5.3.1 Results of Experiment 1

For each instance in benchmark, run it five times and record its best makespan and average makespan. The parameters of GA used here are set as follows: the population size = 20; the crossover probability = 0.9; the mutation probability = 0.5; the maximum number of iterations = 500. The computational results are presented in Tables 3~7. Among them, “Mode3” represents the solution representation proposed in Section 3.

| Job | Stage | AGV | α   | Mode1 Best | Mode1 Average | Mode2 Best | Mode2 Average | Mode3 Best | Mode3 Average |
|-----|-------|-----|-----|----------|---------------|----------|---------------|----------|---------------|
| 10  | 5     | 2   | 0.10| 612      | 622           | 534      | 591           | 424      | 435           |
|     |       | 4   |     | 508      | 512           | 526      | 577           | 423      | 434           |
|     | 6     | 1092| 1111| 1159    | 1227          |          |               | 832      | 854           |
|     | 2     | 449 | 458 | 514      | 563           |          |               | 423      | 431           |
|     | 4     | 951 | 967 | 1107    | 1157          |          |               | 830      | 845           |
|     | 6     | 888 | 905 | 1085    | 1145          |          |               | 835      | 843           |
|     | 10    | 2   |     | 1176    | 1183          |          |               | 688      | 707           |
|     |       | 5   | 1058| 1068    | 938           | 988      |              | 686      | 698           |
|     | 4     | 925 | 949 | 894     | 959           |          |               | 688      | 698           |
|     | 6     | 1812| 1841| 1703    | 1767          |          |               | 974      | 1007          |
|     | 10    | 2   |     | 1574    | 1596          |          |               | 871      | 892           |
|     |       | 5   | 1422| 1443    | 1472          | 1495     |              | 867      | 885           |
|     | 4     | 1481| 1488| 1228    | 1273          |          |               | 742      | 758           |
|     | 6     | 1377| 1391| 1126    | 1184          |          |               | 719      | 729           |
|     | 10    | 2   |     | 1276    | 1291          |          |               | 1485     | 1510          |
|     |       | 5   | 2959 | 2966    | 2443          | 2508     |              | 1149     | 1197          |
|     | 4     | 2720| 2729| 2217    | 2382          |          |               | 1151     | 1176          |
|     | 6     | 2353| 2537| 2213    | 2301          |          |               |          |               |

| Job | Stage | AGV | α   | Mode1 Best | Mode1 Average | Mode2 Best | Mode2 Average | Mode3 Best | Mode3 Average |
|-----|-------|-----|-----|----------|---------------|----------|---------------|----------|---------------|
| 10  | 5     | 2   | 0.41| 736      | 757           | 763      | 813           | 611      | 635           |
|     |       | 4   |     | 578      | 589           | 655      | 698           | 471      | 484           |
|     | 6     | 509 | 520 | 560     | 643           |          |               | 469      | 478           |
|     | 10    | 2   | 0.60| 1412     | 1430          |          |               | 1255     | 1291          |
processing and transportation of jobs are carried out depending on the stage (the order sequence at the latter stages is determined by the end time of jobs at the previous stage). This will make each AGV only transport jobs with the same stage in a period of time. As a result, the scheduling schemes obtained with this mode are too limited; (2) For Mode2, under the same scale of problem, the coding length is much larger than that of others. On the one hand, it can almost represent all the solutions in the solution space, providing the possibility for the algorithm to find global optimal solutions. On the other hand, due to the large search range, it is difficult for the algorithm to

Table 5  Comparison on Group 1 with large-size and small-α

| Job | Stage | AGV | α  | Mode1 | Mode2 | Mode3 |
|-----|-------|-----|----|-------|-------|-------|
|     |       |     |    | Best  | Average | Best | Average | Best  | Average |
| 80  | 5     | 6   | 0.10 | 4449  | 4474  | 3246 | 3321 | 2066  | 2055  |
|     | 8     | 4   | 0.10 | 4337  | 4347  | 3122 | 3255 | 2014  | 2053  |
| 160 | 5     | 6   | 0.10 | 9661  | 9684  | 7305 | 7514 | 4265  | 4291  |
|     | 8     | 4   | 0.10 | 9537  | 9544  | 7097 | 7302 | 4234  | 4268  |

Table 6  Comparison on Group 1 with large-size and large-α

| Job | Stage | AGV | α  | Mode1 | Mode2 | Mode3 |
|-----|-------|-----|----|-------|-------|-------|
|     |       |     |    | Best  | Average | Best | Average | Best  | Average |
| 80  | 5     | 6   | 0.43 | 4544  | 4552  | 4086 | 4212 | 3078  | 3093  |
|     | 8     | 4   | 0.43 | 4440  | 4441  | 3793 | 3800 | 2288  | 2308  |
| 160 | 5     | 6   | 0.40 | 9859  | 9876  | 8435 | 8545 | 6444  | 6463  |
|     | 8     | 4   | 0.40 | 9753  | 9773  | 7687 | 7737 | 4871  | 4926  |

Table 7  Comparison on Group 2

| Job | Stage | AGV | α  | Mode1 | Mode2 | Mode3 |
|-----|-------|-----|----|-------|-------|-------|
|     |       |     |    | Best  | Average | Best | Average | Best  | Average |
| 2   | 4     | 0.80 | 393  | 400  | 448  | 475  | 387  | 397  |
| 4   | 6     | 0.73 | 499  | 504  | 656  | 671  | 485  | 493  |
| 10  | 6     | 0.71 | 756  | 771  | 1013 | 1043 | 729  | 735  |
| 8   | 10    | 0.69 | 901  | 911  | 1273 | 1288 | 835  | 846  |
| 10  | 10    | 0.69 | 1092 | 1108 | 1535 | 1639 | 1058 | 1062 |
| 2   | 6     | 0.77 | 578  | 584  | 590  | 633  | 496  | 502  |
| 4   | 8     | 0.65 | 923  | 930  | 1081 | 1129 | 746  | 766  |
| 20  | 6     | 0.67 | 1112 | 1121 | 1415 | 1530 | 905  | 910  |
| 8   | 12    | 0.70 | 1494 | 1504 | 1965 | 2046 | 1241 | 1245 |
| 10  | 14    | 0.67 | 1733 | 1748 | 2309 | 2388 | 1336 | 1364 |

In the two groups of instances shown in Tables 3–7, the solution representation (Mode3) proposed in this paper get all optimal solutions in best makespan and average makespan, which is much better than others.
find a satisfactory solution in a limited time; (3) For Mode3, by optimizing the order sequence at the first stage that has a greater impact on the makespan, and combining with effective heuristic rule, the solution representation performs better.

The above results show that the proposed solution representation can help algorithm obtaining more and better results than others, under the same number of iterations and other parameters of the algorithm used.

Figure 6  Gantt charts of three different solution representations

5.3.2 Results of Experiment 2
For each instance in benchmark, run it five times and record its best makespan and average makespan. The parameters of GATS used here are set as follows: the population size = 20; the crossover probability = 0.9; the mutation probability = 0.5; the length of tabu list = 10; the maximum iterations of the hybrid GA and TS = 50; the maximum iterations of the TS is 20. The computational results are presented in Tables 8~12. Among them, “GATS_FCFS” represents the hybrid GA and TS with FCFS rule.

Table 8  Comparison on Group 1 with small-size and small-α

| Job | Stage | AGV | α  | GA | TS_FCFS | GA | TS_LTBA | GA | TS_STD | GA | TS_GWTQ |
|-----|-------|-----|----|----|---------|----|---------|----|---------|----|---------|
|     |       |     |    | Best | Average |     | Best | Average |     | Best | Average |     |
| 10  | 2     | 5   | 0.10 | 424 | 426     | 423 | 422 | 422 | 456 | 467 | 423 | 424 |
|     | 6     |     |     | 423 | 423     | 423 | 423 | 433 | 442 | 423 | 423 |
| 10  | 2     | 4   | 0.14 | 830 | 833     | 842 | 844 | 948 | 996 | 830 | 845 |
|     | 6     |     |     | 830 | 835     | 830 | 833 | 926 | 967 | 830 | 840 |
| 20  | 5     | 4   | 0.10 | 687 | 694     | 689 | 698 | 820 | 837 | 686 | 696 |
|     | 6     |     |     | 679 | 685     | 677 | 682 | 779 | 799 | 675 | 681 |
| 10  | 4     | 0.17 | | 974 | 991     | 1024 | 1038 | 1166 | 1190 | 975 | 992 |
|     | 6     |     |     | 860 | 861     | 861 | 869 | 1074 | 1093 | 851 | 859 |
| 30  | 5     | 6   | 0.11 | 686 | 699     | 694 | 702 | 838 | 849 | 692 | 702 |
|     | 2     | 1485 | 1531 | 1590 | 1603 | 1500 | 1602 | 1510 | 1532 | 1135 | 1154 |
| 10  | 6     | 1115 | 1123 | 1132 | 1144 | 1385 | 1395 | 1111 | 1127 |     |     |

Number of optimal solutions 10 9 7 6 0 0 10 6

Table 9  Comparison on Group 1 with small-size and large-α

| Job | Stage | AGV | α  | GA | TS_FCFS | GA | TS_LTBA | GA | TS_STD | GA | TS_GWTQ |
|-----|-------|-----|----|----|---------|----|---------|----|---------|----|---------|
|     |       |     |    | Best | Average |     | Best | Average |     | Best | Average |     |
| 10  | 2     | 5   | 0.41 | 604 | 615     | 615 | 640 | 644 | 664 | 596 | 613 |
|     | 6     |     |     | 469 | 471     | 471 | 475 | 504 | 519 | 467 | 471 |
| 10  | 4     | 2   | 0.60 | 1250 | 1278 | 1330 | 1345 | 1416 | 1451 | 1276 | 1302 |
|     | 6     |     |     | 938 | 944     | 950 | 967 | 1043 | 1108 | 929 | 938 |
| 10  | 4     | 6   |     | 905 | 913     | 911 | 920 | 983 | 1030 | 900 | 909 |
In each type of instances, the statistics of the total number that each rule can achieve optimal result about best makespan and average makespan among the four are shown in the Tables 13–14. In the 58 instances, the GATS with FCFS rule achieves 30 optimal results about best makespan and 29 optimal results about average makespan, which is better than other three. It is evident that the FCFS rule used in solution evaluation is effective and reliable when solving integrated scheduling problem.

Table 10 Comparison on Group 1 with large-size and small-α

| Job | Stage | AGV | α  | GATS_FCFS | GATS_LTBA | GATS_STD | GATS_GWTQ |
|-----|-------|-----|----|-----------|-----------|----------|-----------|
|     |       |     |    | Best      | Average   | Best      | Average   | Best      | Average   |
| 80  | 5     | 4   | 0.10 | 2024 | 2008     | 1995     | 2005     | 2300     | 2306     |
|     | 6     | 8   | 0.10 | 1996 | 2008     | 1995     | 2005     | 2300     | 2306     |
|     | 6     | 4   | 0.10 | 4212 | 4241     | 4238     | 4272     | 4803     | 4817     |
| 160 | 5     | 6   | 0.10 | 4195 | 4213     | 4205     | 4226     | 4680     | 4720     |
|     | 6     | 8   | 0.10 | 4199 | 4213     | 4233     | 4237     | 4671     | 4694     |

Table 11 Comparison on Group 1 with large-size and large-α

| Job | Stage | AGV | α  | GATS_FCFS | GATS_LTBA | GATS_STD | GATS_GWTQ |
|-----|-------|-----|----|-----------|-----------|----------|-----------|
|     |       |     |    | Best      | Average   | Best      | Average   | Best      | Average   |
| 80  | 5     | 4   | 0.43 | 3022 | 3039     | 3212     | 3231     | 2732     | 2747     |
|     | 6     | 8   | 0.43 | 2076 | 2085     | 2068     | 2099     | 2324     | 2381     |
|     | 6     | 4   | 0.43 | 6380 | 6414     | 6771     | 6774     | 6450     | 6453     |
| 160 | 5     | 6   | 0.40 | 4681 | 4701     | 5071     | 5072     | 5002     | 5027     |
|     | 6     | 8   | 0.40 | 4373 | 4416     | 4385     | 4453     | 4902     | 4924     |

Table 12 Comparison on Group 2

| Job | Stage | AGV | α  | GATS_FCFS | GATS_LTBA | GATS_STD | GATS_GWTQ |
|-----|-------|-----|----|-----------|-----------|----------|-----------|
|     |       |     |    | Best      | Average   | Best      | Average   | Best      | Average   |
| 2   | 4     | 0.80 | 377 | 380     | 381       | 387       | 379       | 381       | 378       |
| 4   | 6     | 0.73 | 474 | 479     | 491       | 494       | 475       | 486       | 471       |
| 10  | 6     | 0.71 | 712 | 731     | 738       | 755       | 734       | 752       | 717       |
| 8   | 10    | 0.69 | 832 | 838     | 842       | 846       | 846       | 860       | 838       |
| 10  | 10    | 0.69 | 1038| 1050    | 1045      | 1056      | 1084      | 1106      | 1050      |
| 2   | 6     | 0.77 | 465 | 481     | 484       | 489       | 504       | 513       | 473       |
| 4   | 8     | 0.65 | 740 | 755     | 788       | 799       | 767       | 795       | 732       |
| 20  | 6     | 0.67 | 885 | 899     | 915       | 924       | 977       | 994       | 888       |
| 8   | 12    | 0.70 | 1216| 1229    | 1277      | 1285      | 1267      | 1290      | 1221      |
| 10  | 14    | 0.67 | 1348| 1369    | 1404      | 1424      | 1428      | 1439      | 1374      |

Table 13 Statistical result of the best makespan

| Size | α  | Total number | Number of optimal solutions |
|------|----|--------------|-----------------------------|
| small| small| 18 | 10 | 7 | 0 | 10 |
| small| large| 18 | 5 | 2 | 4 | 5 | 7 |
| large| small| 6 | 3 | 1 | 0 | 3 |
| large| large| 6 | 4 | 1 | 1 | 0 |
| Group2 | 10 | 8 | 0 | 0 | 2 |
| Total  | 58 | 30 | 11 | 6 | 22 |
An integrated production and transportation scheduling method in hybrid flow shop

### Table 14 Statistical result of the average makespan

| Size | α  | Total number | GATS_F | GATS_LTBA | GATS_STD | GATS_WTQ |
|------|----|--------------|--------|-----------|----------|----------|
| small | small | 18 | 9 | 6 | 0 | 6 |
| small | large | 18 | 8 | 1 | 4 | 8 |
| large | small | 6 | 3 | 1 | 0 | 2 |
| large | large | 6 | 3 | 0 | 1 | 2 |
| Group2 | 10 | 6 | 0 | 0 | 4 |
| Total | 58 | 29 | 8 | 5 | 22 |

### Table 15 Comparison about best makespan on Group 1 with small-size and small-α

| Job | Stage | AGV | α | GA | SA | ABC | GWO | MBO | GATS |
|-----|-------|-----|----|----|----|-----|-----|-----|------|
| 2   | 5     | 0.10 | 424 | 425 | 421 | 423 | 421 | 421 |
| 6   | 4     | 0.14 | 830 | 830 | 830 | 830 | 830 | 830 |
| 2   | 5     | 0.11 | 719 | 725 | 734 | 734 | 734 | 734 |
| 6   | 6     | 0.16 | 1149 | 1149 | 1149 | 1149 | 1149 | 1149 |

### Table 16 Comparison about average makespan on Group 1 with small-size and small-α

| Job | Stage | AGV | α | GA | SA | ABC | GWO | MBO | GATS |
|-----|-------|-----|----|----|----|-----|-----|-----|------|
| 2   | 5     | 0.10 | 435 | 433 | 425 | 425 | 425 | 425 |
| 6   | 4     | 0.14 | 845 | 844 | 843 | 843 | 843 | 843 |
| 2   | 5     | 0.11 | 724 | 724 | 733 | 733 | 733 | 733 |
| 6   | 6     | 0.16 | 1176 | 1176 | 1176 | 1176 | 1176 | 1176 |

### Table 17 Comparison about best makespan on Group 1 with small-size and large-α

| Job | Stage | AGV | α | GA | SA | ABC | GWO | MBO | GATS |
|-----|-------|-----|----|----|----|-----|-----|-----|------|
| 2   | 5     | 0.41 | 611 | 610 | 593 | 613 | 613 | 613 |
| 6   | 4     | 0.60 | 1255 | 1254 | 1242 | 1259 | 1226 | 1226 |
| 2   | 5     | 0.38 | 797 | 792 | 796 | 802 | 796 | 796 |

### 5.3.3 Results of Experiment 3

For each instance, run each algorithm five times and record its best makespan and average makespan. The computational results are presented in Tables 15–24.
Table 18  Comparison about average makespan on Group 1 with small-size and large-α

| Job | Stage | AGV | α  | GA   | SA   | ABC  | ABC  | GWO  | MBO  | GATS |
|-----|-------|-----|-----|------|------|------|------|------|------|------|
| 6   | 2     | 0.72| 2   | 0.72 | 2    | 0.72 | 2    | 0.72 | 2    | 0.72 |
| 10  | 0.45  | 1067| 0.45| 1067 | 0.45 | 1067 | 0.45 | 1067 | 0.45 | 1067 |
| 30  | 0.67  | 3998| 0.67| 3998 | 0.67 | 3998 | 0.67 | 3998 | 0.67 | 3998 |

Number of optimal solutions 5 2 1 0 5 9

Table 19  Comparison about best makespan on Group 1 with large-size and small-α

| Job | Stage | AGV | α  | GA   | SA   | ABC  | ABC  | GWO  | MBO  | GATS |
|-----|-------|-----|-----|------|------|------|------|------|------|------|
| 6   | 4     | 0.10| 4   | 0.10 | 4    | 0.10 | 4    | 0.10 | 4    | 0.10 |
| 8   | 4     | 0.10| 4   | 0.10 | 4    | 0.10 | 4    | 0.10 | 4    | 0.10 |
| 160 | 4     | 0.10| 4   | 0.10 | 4    | 0.10 | 4    | 0.10 | 4    | 0.10 |

Number of optimal solutions 0 1 0 0 0 5

Table 20  Comparison about average makespan on Group 1 with large-size and small-α

| Job | Stage | AGV | α  | GA   | SA   | ABC  | ABC  | GWO  | MBO  | GATS |
|-----|-------|-----|-----|------|------|------|------|------|------|------|
| 6   | 4     | 0.43| 4   | 0.43 | 4    | 0.43 | 4    | 0.43 | 4    | 0.43 |
| 8   | 4     | 0.43| 4   | 0.43 | 4    | 0.43 | 4    | 0.43 | 4    | 0.43 |
| 160 | 4     | 0.43| 4   | 0.43 | 4    | 0.43 | 4    | 0.43 | 4    | 0.43 |

Number of optimal solutions 0 0 0 0 0 5

Table 21  Comparison about best makespan on Group 1 with large-size and large-α

| Job | Stage | AGV | α  | GA   | SA   | ABC  | ABC  | GWO  | MBO  | GATS |
|-----|-------|-----|-----|------|------|------|------|------|------|------|
| 6   | 4     | 0.43| 4   | 0.43 | 4    | 0.43 | 4    | 0.43 | 4    | 0.43 |
| 8   | 4     | 0.43| 4   | 0.43 | 4    | 0.43 | 4    | 0.43 | 4    | 0.43 |
| 160 | 4     | 0.43| 4   | 0.43 | 4    | 0.43 | 4    | 0.43 | 4    | 0.43 |

Number of optimal solutions 0 0 0 0 0 5
Table 22  Comparison about average makespan on Group 1 with large-size and large-α

| Job | Stage | AGV | α  | GA | SA | ABC | GWO | MBO | GATS |
|-----|-------|-----|----|----|----|-----|-----|-----|------|
| 80  | 5     | 6   | 0.43 | 3093 | 3093 | 3077 | 3079 | 3074 | 3039 |
|     | 8     | 4   | 6463 | 6504 | 6501 | 6471 | 6515 | 6414 |
| 160 | 5     | 6   | 0.40 | 4926 | 4812 | 5029 | 4799 | 4813 | 4701 |
|     | 8     | 4   | 4567 | 4430 | 4663 | 4394 | 4594 | 4416 |

Number of optimal solutions: 0 0 0 1 0 5

Table 23  Comparison about best makespan on Group 2

| Job | Stage | AGV | α  | GA | SA | ABC | GWO | MBO | GATS |
|-----|-------|-----|----|----|----|-----|-----|-----|------|
| 2   | 4     | 0.80 | 387 | 384 | 378 | 393 | 381 | 377 |
| 4   | 6     | 0.73 | 485 | 479 | 472 | 490 | 479 | 474 |
| 10  | 6     | 0.71 | 728 | 728 | 717 | 724 | 718 | 712 |
| 8   | 10    | 0.69 | 835 | 832 | 845 | 848 | 834 | 832 |
| 10  | 10    | 0.69 | 1058 | 1047 | 1053 | 1058 | 1051 | 1038 |
| 2   | 6     | 0.77 | 496 | 494 | 497 | 497 | 479 | 465 |
| 4   | 8     | 0.65 | 746 | 751 | 751 | 750 | 740 | 740 |
| 20  | 6     | 0.67 | 895 | 911 | 906 | 914 | 894 | 885 |
| 8   | 12    | 0.70 | 1241 | 1230 | 1222 | 1240 | 1203 | 1216 |
| 10  | 14    | 0.67 | 1336 | 1365 | 1362 | 1363 | 1336 | 1336 |

Number of optimal solutions: 1 1 1 0 0 3 8

Table 24  Comparison about average makespan on Group 2

| Job | Stage | AGV | α  | GA | SA | ABC | GWO | MBO | GATS |
|-----|-------|-----|----|----|----|-----|-----|-----|------|
| 2   | 4     | 0.80 | 397 | 389 | 384 | 396 | 383 | 380 |
| 4   | 6     | 0.73 | 493 | 482 | 480 | 492 | 481 | 479 |
| 10  | 6     | 0.71 | 735 | 733 | 725 | 739 | 722 | 731 |
| 8   | 10    | 0.69 | 846 | 843 | 847 | 852 | 837 | 838 |
| 10  | 10    | 0.69 | 1062 | 1060 | 1057 | 1060 | 1053 | 1050 |
| 2   | 6     | 0.77 | 502 | 500 | 502 | 526 | 484 | 481 |
| 4   | 8     | 0.65 | 766 | 763 | 762 | 775 | 746 | 755 |
| 20  | 6     | 0.67 | 910 | 920 | 920 | 925 | 900 | 899 |
| 8   | 12    | 0.70 | 1245 | 1238 | 1233 | 1247 | 1214 | 1229 |
| 10  | 14    | 0.67 | 1364 | 1371 | 1364 | 1372 | 1344 | 1369 |

Number of optimal solutions: 0 0 0 0 5 5

In each type of instances, the statistics of the total number that each algorithms can achieve optimal results about best makespan and average makespan among the six are shown in the Tables 25~26. In the 58 instances, the GATS achieves 40 optimal results about best makespan and 38 optimal results about average makespan, which is better than other five. This means that the hybrid GA and TS has both effectiveness and efficiency for solving integrated scheduling problem.

Table 25  Statistical result of best makespan

| Size | α    | Total number | GA | SA | ABC | GWO | MBO | GATS |
|------|------|--------------|----|----|-----|-----|-----|------|
| small | small | 18           | 3 | 1 | 4 | 2 | 12 | 13 |
| small | large | 18           | 5 | 2 | 1 | 0 | 5 | 9 |
| large | small | 6            | 0 | 1 | 0 | 0 | 0 | 5 |
| large | large | 6            | 0 | 0 | 0 | 1 | 0 | 5 |
| Group 2 |       | 10           | 1 | 1 | 1 | 0 | 3 | 8 |
| Total |       | 58           | 9 | 5 | 6 | 3 | 20 | 40 |

Table 26  Statistical result of average makespan

| Size | α    | Total number | GA | SA | ABC | GWO | MBO | GATS |
|------|------|--------------|----|----|-----|-----|-----|------|
| small | small | 18           | 3 | 1 | 4 | 2 | 12 | 13 |
| small | large | 18           | 5 | 2 | 1 | 0 | 5 | 9 |
| large | small | 6            | 0 | 1 | 0 | 0 | 0 | 5 |
| large | large | 6            | 0 | 0 | 0 | 1 | 0 | 5 |
| Group 2 |       | 10           | 1 | 1 | 1 | 0 | 3 | 8 |
| Total |       | 58           | 9 | 5 | 6 | 3 | 20 | 40 |
6 Conclusions and future work

This paper proposes an integrated scheduling method for production and transportation in hybrid flow shop environment. Firstly, some preparation for the problem has been made. Then, a hybrid genetic algorithm with tabu search has been adapted. Finally, two groups of instances have been designed and three types of experiments have been carried out for verifying the effectiveness of the proposed method.

The contributions of this research include:

- A new solving method for IPTSP, including the establishment of task pool, the new solution representation and the new solution evaluation has been proposed for the problem. Taking it as coding and decoding method, the algorithm can search for a satisfactory solution within appropriate time;
- A hybrid algorithm which hybridizes the genetic algorithm and tabu search has been proposed to solve the integrated scheduling problem. The hybrid algorithm combines the advantages of the two algorithms. The experimental results show that this algorithm has both effectiveness and efficiency for solving integrated scheduling problem.

Although the method proposed in this paper has achieved good results, there are still some works can be made in the future. Firstly, we can use multiple rules instead of single rule in solution evaluation, which may make the results more stable. Secondly, we can design some efficient methods of population initialization instead of the random one, and design efficient neighborhood structure to improve the hybrid algorithm.

7 Declaration

Acknowledgements
Not applicable

Funding
Supported by the National Key R&D Program of China (Grant no. 2019YFB1704603) and the National Natural Science Foundation of China (Grant no. 51775216).

Availability of data and materials

The datasets supporting the conclusions of this article can be found at the website [29].

Authors’ contributions
The author contributions are as follows: Wangming Li was in charge of the whole trial; Wangming Li wrote the manuscript; Xinyu Li assisted with problem analyses. All authors read and approved the final manuscript.

Competing interests
The authors declare no competing financial interests.

Consent for publication
Not applicable

Ethics approval and consent to participate
Not applicable

References
[1] Wei Qin, Zilong Zhuang, Yang Liu, et al. A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly. Computers & Industrial Engineering, 2019, 138.
[2] Leilei Meng, Chaoyong Zhang, Xinyu Shao, et al. More MILP models for hybrid flow shop scheduling problem and its extended problems. International journal of production research, 2020, 58(13): 3905-3930.
[3] Quan-ke Pan, Ling Wang, Kun Mao, et al. An Effective Artificial Bee Colony Algorithm for a Real-World Hybrid Flowshop Problem in Steelmaking Process. IEEE Transactions on Automation Science and Engineering, 2013, 10(2): 307-322.
[4] Biao Zhang, Quan-ke Pan, Liang Gao, et al. A Three-Stage Multiobjective Approach Based on Decomposition for an Energy-Efficient Hybrid Flow Shop Scheduling Problem. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(12): 4984-4999.
[5] Xingyu Li, Baicun Wang, Chao Liu, et al. Intelligent Manufacturing Systems in COVID-19 Pandemic and Beyond: Framework and Impact Assessment. Chinese Journal of Mechanical Engineering, 2020, 33(1).
[6] Ning Zhao, Song Ye, Kaidian Li, et al. Effective Iterated Greedy Algorithm for Flow-Shop Scheduling Problems with Time lags. Chinese Journal of Mechanical Engineering, 2017, 30(3): 652-662.
[7] Lin Gui, Liang Gao, Xinyu Li. Anomalies in Special Permutation Flow Shop Scheduling Problems. Chinese Journal of Mechanical Engineering, 2020, 33(1).
[8] Ning Zhao, Siyu Chen, Yanhua Du. Emergency local searching approach for job shop scheduling. Chinese Journal of Mechanical Engineering, 2013, 26(5): 918-927.
An integrated production and transportation scheduling method in hybrid flow shop

[9] Guomin Li, Xinyu Li, Liang Gao, et al. Tasks assigning and sequencing of multiple AGVs based on an improved harmony search algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 2019, 10(11): 4533-4546.

[10] T Miyamoto, K Inoue. Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Computers & Industrial Engineering*, 2016, 91: 1-9.

[11] Dunbing Tang, Min Dai. Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem. *Chinese Journal of Mechanical Engineering*, 2015, 28(5): 1048-1055.

[12] G Ulusoy, Ü Bilge. Simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, 1993, 31(12):2857-2873.

[13] A Ahmadi-Javid, P Hoooshangi-Tabrizi. A mathematical formulation and anarchic society optimisation algorithms for integrated scheduling of processing and transportation operations in a flow-shop environment. *International Journal of Production Research*, 2015, 53(19): 5988-6006.

[14] T Nishi, Y Hiranaka, I E Grossmann. A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Computers & Operations Research*, 2011, 38(5): 876-888.

[15] A Elmi, S Topaloglu. A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots. *Computers & Operations Research*, 2013, 40(10): 2543-2555.

[16] S Zabihzadeh, J Rezaeian. Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time. *Applied Soft Computing*, 2016, 40: 319-330.

[17] M K Marichelvam, T Prabaharan, Yang X S. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Applied Soft Computing*, 2014, 19: 93-101.

[18] Ghiahi Al Aqel, Xinyu Li, Liang Gao. A Modified Iterated Greedy Algorithm for Flexible Job Shop Scheduling Problem. *Chinese Journal of Mechanical Engineering*, 2019, 32(1).

[19] Y Ho, H Liu. The performance of load-selection rules and pickup-dispatching rules for multiple-load AGVs. *Journal of Manufacturing Systems*, 2009, 28(1): 1-10.

[20] S A Brah, L L Luan. Heuristics for scheduling in a flow shop with multiple processors. *European Journal of Operational Research*, 1999, 113(1): 113-122.

[21] F Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 1986, 13: 533-549.

[22] Victor Fernandez-Viagas, Josem Framinan. Design of a testbed for hybrid flow shop scheduling with identical machines. *Computers & Industrial Engineering*, 2020, 141.

[23] G Ulusoy, F Sivrikaya-serifoğlu, Ü Bilge. A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, 1997, 24(4): 335-351.

[24] Yingli Li, Xinyu Li, Liang Gao, et al. An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times. *Computers & Industrial Engineering*, 2020, 147.

[25] S Mirjalili, S M Mirjalili, A Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69: 46-61.

[26] E Duman, M Uysal, A F Alkaya. Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem. *Information Sciences*, 2012, 217: 65-77.

[27] F S Şerifoğlu, G Ulusoy. Multiprocessor task scheduling in multistage hybrid flowshops: a genetic algorithm approach. *Journal of the Operational Research Society*, 2004, 55(5): 504-512.

[28] James Lin, Chun-chih Chiu, Yu-hsiang Chang. Simulation-based optimization approach for simultaneous scheduling of vehicles and machines with processing time uncertainty in FMS. *Flexible Services and Manufacturing Journal*, 2019, 31(1): 104-141.

[29] Two groups of instances for IPTSP. https://github.com/WaminLee/Instances-For-Integrated-Scheduling-P

Biographical notes

Wangming Li, born in 1997, is currently a master candidate at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his bachelor degree from Huazhong University of Science and Technology, China, in 2019. His main research interests are scheduling problem and algorithm optimization.

Dong Han, born in 1997, is currently a master candidate at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. She received his bachelor degree from Huazhong University of Science and Technology, China, in 2019. Her main research interests are deep learning and algorithm optimization.

Liang Gao, born in 1974, is currently a professor at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his PhD degree in mechatronic engineering from Huazhong University of Science and Technology, China, in 2002. His main research interests are intelligent optimization method and its application in design and manufacturing.

Xinyu Li, born in 1985, is currently a professor at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his PhD degree in industrial engineering from Huazhong University of Science and Technology, China, in 2009. His main research interests are intelligent manufacturing systems, shop scheduling, intelligent optimization and machine learning.

Yang Li, born in 1996, is currently a doctor at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China. He received his bachelor degree from Huazhong University of Science and Technology, China, in 2018. His main research interests are flow shop scheduling and algorithm optimization.
Figures

Figure 1
An example for the integrated scheduling problem

Figure 2
Simple description of the task pool
Figure 3

Illustration of the transport sequence vector

| Position: Priority $u$ | 1 | 2 | 3 | 4 | 5 |
|------------------------|---|---|---|---|---|
| Task Indicated         | Task3 Task2 Task1 Task4 Task5 |
| Transport sequence $v_1(u)$ | 3 | 2 | 1 | 4 | 5 |

Figure 4

Illustration of the transporter assignment vector

| Position: $v$ | 1 | 2 | 3 | 4 | 5 |
|---------------|---|---|---|---|---|
| Task Indicated| Task1 Task2 Task3 Task4 Task5 |
| Transporter Assignment $v_1(v)$ | 2 | 2 | 1 | 2 | 1 |
Figure 5

Framework of the method

Figure 6

(a) Gantt chart of Model1  (b) Gantt chart of Model2  (c) Gantt chart of Model3
Gantt charts of three different solution representations