Dynamic Integrated Flexible Job Shop Scheduling with Transportation Robot

Yingmei He[1], Bin Xin[1,*], Sai Lu[1], and Yulong Ding[2]

[1]School of Automation, Beijing Institute of Technology, Beijing, China
E-mail: brucebin@bit.edu.cn (Corresponding author)
[2] Peng Cheng Laboratory, Shenzhen, China

Abstract. The integrated scheduling problem for machines and transportation robots in flexible job shop is a complex but valuable problem. In this paper, the dynamic integrated scheduling problem considering breakdowns, order insertions and battery consumption of robots is studied, which is aiming at minimizing the order completion time (makespan). Adopting an event-driven global rescheduling strategy, three algorithms are designed, including the Genetic Algorithm (GA), the Improved Variable Neighborhood Search (IVNS) algorithm, and the Memetic Algorithm (MA). Finally, numerical experiments are carried out to test the performance of the algorithms. The effectiveness of the rescheduling strategy is verified under dynamic situations.

Keywords: Dynamic Scheduling, Flexible Job Shop, Multi-agent, Charging

1. INTRODUCTION

At present, the large-scale standardized production mode is changing to the direction of small-batch and customized mode. The change of manufacturing mode makes the production process complex and changeable, which requires job shops to be more flexible. In addition to developing the flexibility of machines (There is more than one eligible machine for each process), the transportation system has the ability to improve the flexibility of manufacturing system. More and more production systems use Automated Guided Vehicles (AGVs) as the transfer tools of materials. Because of the strong coupling between the machining process and the transportation process, it is necessary to integrate the scheduling of the machines and the AGVs. Moreover, all kinds of unexpected situations may occur during the production at any time, which must be dealt with in time to prevent affecting the production progress. Therefore, solving the Dynamic Integrated Flexible Job Shop Scheduling Problem (DIFJSP) is the key to improve the production efficiency and reduce the operation cost.

Scholars at home and abroad have done some research on the Integrated Flexible Job Shop Scheduling Problem (IFJSP) which considers transportation time. Tan *et al.* [1] proposed an enhanced Nondominated Sorting Genetic Algorithm-II (NSGA-II) for the IFJSP with the objective of minimizing the worker fatigue and the makespan. Homayouni *et al.* [2] proposed a mixed integer linear programming model and a heuristic algorithm based on local search for the IFJSP. Li *et al.* [3] developed an imperialist competitive algorithm with feedback for the IFJSP, considering sequence-dependent setup times and energy efficiency. Yan *et al.* [4] developed a Genetic Algorithm (GA) and proposed an entity-JavaScript object notation method for the IFJSP. Wang *et al.* [5] proposed a multi-objective optimization algorithm with multi-region division sampling strategy considering preventive maintenance to solve the IFJSP.

For the Dynamic Flexible Job Shop Scheduling Problem (DFJSP), Luo *et al.* [6] proposed an on-line rescheduling framework with new job insertions. Xu *et al.* [7] proposed a heuristic template with three different delayed routing strategies based on genetic programming hyper-heuristic for the DFJSP. Zhang *et al.* [8] proposed an improved heuristic Kalman algorithm with the relative position index encoding to solve the DFJSP. Luo *et al.* [9] proposed a double loop deep Q-network method for the DFJSP with random job arrivals. Fan *et al.* [10] presented a mathematical programming model with extended technical precedence constraints and developed a genetic programming to solve the DFJSP.

In order to improve productivity and enhance resource utilization, it is necessary to achieve good coordination between the machining process and the transportation process, and to deal with emergencies in time. Therefore, this paper focuses on dynamic situations of integrated flexible job shop with considering battery capacity constraints and charging demand of AGVs. For the DIFJSP, this paper proposes a Memetic Algorithm (MA) based on the hybrid method of an improved GA and a Variable Neighborhood Search (VNS) algorithm.

The rest of this paper is structured as follows. In Section 2, a mathematical model of the DIFJSP is constructed aiming at minimizing the makespan. In Section 3, the GA, the Improved Variable Neighborhood Search (IVNS) algorithm and the MA are designed for solving the proposed problem. Section 4 shows the computational and comparison results. Section 5 concludes the paper.

2. PROBLEM FORMULATION

2.1. Problem Description

The elements of the job shop are illustrated in Fig. 1. There is a set of n jobs $J=\{J_1,J_2,\ldots,J_n\}$, a set of m machines $M=\{M_1,M_2,\ldots,M_m\}$ and a set of r AGVs

$V=\{V_1,V_2,\ldots,V_r\}$. The AGVs transport raw materials from the warehouse to the machines and undertake the transportation of the jobs between machines. After all processes of a job are completed, the product is transported back to the warehouse. One transportation task includes two parts: taking the job and sending the job to the next eligible machine. When the remaining battery capacity of a AGV is not enough to complete the next task and reach a charging pile, the AGV will first go to the nearest charging pile for charging and then work.

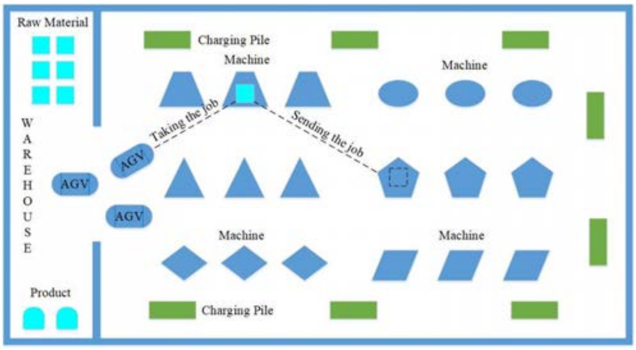


Fig. 1 Flexible Job Shop System with AGVs

This problem also considers the following assumptions:

(1) The sequence of processes about the same job cannot be changed, and there is no sequence constraint between different jobs.

(2) Each machine can only undertake one kind of machining process and only process one job at any time.

(3) Once the process starts, it cannot be interrupted, except in special circumstances, such as machine breakdowns, order cancellation, etc.

(4) Each machine has a job input buffer, a job output buffer and a parking area for the AGVs with sufficient capacity.

(5) At the initial moment, all the jobs and the AGVs are in the warehouse.

(6) Each AGV can only transport one job at a time.

(7) The AGV power consumption rate is related to whether it is loaded or not, and it always maintains a constant speed during operation. After the AGV is fully charged, it can be dispatched from the charging pile position immediately.

(8) The routing conflict problem of AGVs is not considered.

2.2. The Mathematical Model

Based on the above assumptions, the parameters and variables of the job shop model are as follows:

- P_i : the number of operations of the job J_i, $i\in\{1,2,\ldots,n\}$, n is the number of the jobs;
- p_{ij} : the jth process of J_i, $j\in\{1,2,\ldots,P_i\}$;
- k : the machine number of p_{ij}, $k\in\{1,2,\ldots m\}$;
- Ω_{ij} : the set of eligible machines for p_{ij};
- s_{ijk} : the start time of p_{ij} on machine k;
- e_{ijk} : the completion time of p_{ij} on machine k;
- d_{ijk} : the processing time of p_{ij} on machine k;
- I_{ijk} : the time of p_{ij} on the input buffer of machine k;
- Q_{ijk} : the time of p_{ij} on the output buffer of machine k;
- e_i : the processing completion time of J_i;
- ST'_{ijv} : the start time of the empty trip of AGV v for transporting p_{ij}, $v\in\{1,2,\ldots,r\}$;
- ET'_{ijv} : the completion time of the empty trip of AGV v for transporting p_{ij};
- ST_{ijv} : the start time of the loading trip of AGV v for transporting p_{ij};
- ET_{ijv} : the completion time of the loading trip of AGV v for transporting p_{ij};
- t_{ijv} : the transportation time of p_{ij} by AGV v
- v_c : the charging rate of AGV;
- c_0 : the full battery capacity of AGV;
- c_v : the remaining battery capacity of AGV v;
- t_{pv} : the time of AGV v to a charging pile;
- t_{cv} : the total charging time of AGV v;
- $\phi_{ijk}=\begin{cases}1 & \text{if } p_{ij} \text{ is processed on machine } k\\ 0 & \text{otherwise}\end{cases}$
- $\alpha_{i'j'ijk}=\begin{cases}1 & \text{if } p_{i'j'} \text{ is processed on machine } k \text{ before } p_{ij}\\ 0 & \text{otherwise}\end{cases}$
- $\beta_{ijkt}=\begin{cases}1 & \text{if machine } k \text{ is processing } p_{ij} \text{ at time } t\\ 0 & \text{otherwise}\end{cases}$
- $\chi_{ijv}=\begin{cases}1 & \text{if AGV } v \text{ transports } p_{ij}\\ 0 & \text{otherwise}\end{cases}$
- $\gamma_{i'j'ijv}=\begin{cases}1 & \text{if } p_{i'j'} \text{ is transported by AGV } v \text{ before } p_{ij}\\ 0 & \text{otherwise}\end{cases}$
- $\eta_{ijvt}=\begin{cases}1 & \text{if AGV } v \text{ is transporting } p_{ij} \text{ at time } t\\ 0 & \text{otherwise}\end{cases}$
- $\lambda_{ijv}=\begin{cases}1 & \text{if AGV } v \text{ needs to be charged before transporting } p_{ij}\\ 0 & \text{otherwise}\end{cases}$

The goal of scheduling in this paper is to minimize the makespan.

$$T=\min(\max_{i=1}^{n}(e_i)) \tag{1}$$

Subject to

$$e_{ijk}\ge e_{i,j-1,k'}+Q_{i,j-1,k'}+t_{ijv}+I_{ijk}+d_{ijk}, i\in\{1,2,\ldots,n\}, j\in\{2,3,\ldots P_i\}, k,k'\in\{1,2,\ldots,m\}, v\in\{1,2,\ldots,r\} \tag{2}$$

$$e_{i1k'}\ge t_{ijv}+I_{ijk}+d_{ijk},\ i\in\{1,2,\ldots,n\}, j\in\{2,3,\ldots P_i\}, k,k'\in\{1,2,\ldots,m\}, v\in\{1,2,\ldots,r\} \tag{3}$$

$$e_{i,P_i+1,m+1}\ge e_{i,P_i,k'}+Q_{iP_ik'}+t_{iP_iv}, i\in\{1,2,\ldots,n\}, k'\in\{1,2,\ldots,m\}, v\in\{1,2,\ldots,r\} \tag{4}$$

$$e_i\ge e_{i,P_i+1,m+1}, i\in\{1,2,\ldots,n\} \tag{5}$$

$$k=\sum_{k=1}^{m}\phi_{ijk}\Omega_{ij}, i\in\{1,2,\ldots,n\}, j\in\{1,2,\ldots,P_i\} \tag{6}$$

$$\sum_{k=1}^{m}\phi_{ijk}=1, i\in\{1,2,\ldots,n\}, j\in\{1,2,\ldots,P_i\} \tag{7}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{P_i}\beta_{ijkt}\le 1, k\in\{1,2,\ldots,m\}, t\in(0,T) \tag{8}$$

$$e_{ijk}=s_{ijk}+d_{ijk}, i\in\{1,2,\ldots,n\}, j\in\{1,2,\ldots,P_i\}, k\in\{1,2,\ldots,m\} \tag{9}$$

$$s_{ijk}\ge e_{i,j-1,k'}+t_{ijv}+I_{ijk}, i\in\{1,2,\ldots,n\}, j\in\{2,3,\ldots P_i\}, k,k'\in\{1,2,\ldots,m\}, v\in\{1,2,\ldots,r\} \tag{10}$$

$$s_{ijk}+M\left(1-\alpha_{i'j'ijk}\right)\ge e_{i'j'k}, i,i'\in\{1,2,\ldots,n\}, j,j'\in\{1,2,\ldots,P_i\}, k\in\{1,2,\ldots,m\} \tag{11}$$

$$s_{ijk} \ge \max\left\{t_{e\_ijv}, e_{i'j'k}\right\}, i, i' \in \{1,2,\dots,n\}, \quad j, j' \in \{1,2,\dots,P_i\}, k \in \{1,2,\dots,m\}, v \in \{1,2,\dots,r\} \tag{12}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{P_i}\eta_{ijvt} \le 1, \quad v \in \{1,2,\dots,r\}, t \in \left(0,T\right) \tag{13}$$

$$\sum_{v=1}^{r}\chi_{ijv} = 1, i \in \{1,2,\dots,n\}, j \in \{1,2,\dots,P_i\} \tag{14}$$

$$t_{cv} = \left(c_0 - c_v\right)/v_c + t_{pv}, v \in \{1,2,\dots r\} \tag{15}$$

$$ST_{ijv} \ge \max\left\{ET'_{ijv} + \lambda_{ijv}t_{cv}, e_{i,j-1,k'}\right\}, i \in \{1,2\dots,n\}, \quad j \in \{2,3\dots,P_i\}, k \in \{1,2,\dots,m\}, v \in \{1,2,\dots r\} \tag{16}$$

$$ST'_{ijv} + M\left(1-\gamma_{i'j'ijv}\right) \ge ET_{i'j'v}, i, i' \in \{1,2,\dots,n\}, \quad j, j' \in \{1,2,\dots,P_i\}, v \in \{1,2,\dots r\} \tag{17}$$

Constraint (2) ensures the processing sequence between the processes of the same job. Constraint (3) indicates the completion time of the first process. Constraint (4) represents the time for the completed job to be transported back to the warehouse ($P_i + 1$ represents the virtual process of moving back to the warehouse $m+1$). Constraint (5) ensures the completion time of the job is not earlier than the time of arrival in the warehouse. Constraint (6) ensures that the machine for processing the process can only be selected from the eligible machines of this process. Constraint (7) ensures that each process can only be processed on one machine. Constraint (8) ensures that each machine can process at most one job at any time. Constraint (9) ensures that the machining process is continuous. Constraint (10) indicates the starting time of the process. Constraint (11) represents the processing time of adjacent processes on the same machine (M is an infinite positive number). Constraint (12) ensures that the process can start to be processed only when the job arrives and the machine is idle. Constraint (13) ensures that each AGV can only carry one job at any time. Constraint (14) ensures that each transportation task can only be undertaken by one AGV. Constraint (15) indicates the total charging time of the AGV. Constraint (16) ensures that the AGV can start to transport only when it arrives and the preceding process has been finished. Constraint (17) ensures the start time of the task is not earlier than the end time of the preceding task.

2.3. Encoding and Decoding

Based on DIFJSP, the paper adopted a tri-string coding, including the Operation String (OS), the Machine String (MS) and the AGV String (AS) as shown in Fig. 2.

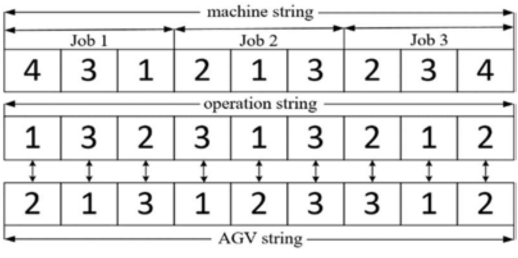


Fig. 2 Tri-string Coding

The OS is based on operation, where each number represents a job number. Since finished jobs need to be transported back to the warehouse, the length of the OS is $L = \sum_{i=1}^{n}\left(P_i + 1\right)$. The lengths of the three kinds of codes are the same. In the AS, each number represents an AGV number and is one-to-one correspondence with the OS. The numbers in the MS represent the machine numbers of processes according to the processing sequence. Taking the second number of the OS as an example, the "3" represents the first process of Job 3 which is transported by AGV 1 and processed on Machine 2.

The decoding process takes the OS as the main line. For each gene of the OS, read the corresponding codes of the MS and the AS, and update the states of the elements. After reading all the codes, the makespan is gotten.

3. Metaheuristic Algorithms for DIFJSP

In metaheuristic algorithms, for the GA, its crossover and mutation operators can easily operate with corresponding constraint rules for the tri-string coding including process sequence, machining and transportation. The VNS can change the search range through neighborhood structures for deep exploitation based on the GA, which has good flexibility and the ability to jump out of the local optimal solution. Therefore, this paper uses the MA which combines the GA and the VNS to solve the proposed problem.

3.1. Dynamic Scheduling Strategy

The dynamic scheduling strategy adopts an event-based global rescheduling. Its core is determining the states of all the job shop elements at the time of emergency and modifying the above states according to the continuity of production to obtain the positions and the unprocessed processes of the jobs, the positions of the AGVs, and the earliest available time of the machines and the AGVs. For example, when an emergency occurs, Machine 1 is processing normally, so its earliest available time is when the process is finished. Then rescheduling is performed with the above states as the initial states.

This paper considers three kinds of dynamic events: machine breakdowns, AGV breakdowns and order insertions, which can occur throughout the makespan.

- Machine breakdowns: The job processed by faulty machine is scrapped. The repair time depends on the degree of the fault. The normal machine in operation will be rescheduled at the current moment or after the current process is completed.
- AGV breakdowns: The faulty AGV in driving will be repaired after being assisted by the staff to complete its transportation task. The normal AGV in driving or during charging will be rescheduled after arriving at the destination or after it is fully charged.
- Order insertions: The inserted jobs are scheduled together with the unfinished jobs of the initial order.

3.2. Genetic Algorithm

The genetic algorithm has good robustness and global search ability. It can find the optimal solution by imitating natural selection and genetic mechanism. The procedure is as follows.

GA procedure

1. Population initialization:
   Randomly generate and evaluate every individual.
2. Iteration:
```
for i from 1 to i_max do
    selection: roulette and elite selection strategy;
    crossover: selecting two chromosomes to cross;
    mutation: exchange and replacement;
end for
return the best individual b
```

Notes: i_{max} - the maximum number of iterations, similarly hereinafter.

3.2.1. Population initialization

The initial population is randomly generated. The OS is generated according to the number of operations of each job. All AGVs are assigned tasks equally in the AS. The MS is generated from the eligible machines of each process to ensure the feasibility of the initial solutions.

3.2.2. Selection

The selection operator adopts the roulette [11] and an improved elite selection strategy. The best individual of each generation is selected to form the elite population with a certain number of individuals. If the number of individuals exceeds the fixed number of the elite population, the worst individual in the elite population will be deleted. New population is generated by roulette and the 10% worst individuals in the new population will be replaced by multiple individuals in the elite population to obtain the next generation population.

3.2.3. Crossover

According to the characteristics of the codes, the three kinds of coding are crossed by corresponding rules, respectively.

- The OS: A precedence operation crossover [12] operator is used to keep the number of operations of each job unchanged. According to the job numbers, the genes are divided into two parts. One part keeps the original position, and the other part adopts the sequence of another chromosome.
- The MS: The crossover operator based on a bit string [13] is adopted to ensure the feasibility of the solutions. Its main idea is to select a specific part of genes and replace them with the genes at the corresponding position in another chromosome.
- The AS: It adopts the same crossover operator as the MS to ensure the variability of the number of transportation tasks undertaken by each AGV.

3.2.4. Mutation

For the MS, a gene is selected and replaced randomly in its eligible machine set. For the OS and the AS, two genes are randomly selected and exchanged.

3.3. Improved Variable Neighborhood Search

In this paper, the IVNS increases a certain number of iterations based on the VNS [14] and uses multiple neighborhood structures to search.

Three kinds of neighborhood structure operators [15] used in the paper are as follows.

- Relocation Search Operator selects a code and inserts it into other position randomly.
- Exchange Search Operator selects two codes and exchanges their positions.
- 2-opt Search Operator selects a piece of code and inserts it into the original position after reverse order.

The above three operators can be used for the OS and the AS. For the MS, select one or more genes, and replace from their eligible processing machine sets. The procedure is as follows.

IVNS procedure

1. Individual initialization:
   Randomly generate an initial individual. Select k_{max} neighborhood structures and the maximum step size l_{max}.
2. Iteration:
```
for i from 1 to i_max do
    for k from 1 to k_max do
        for l from 1 to l_max do
            if find a better individual b' then
                b ← b', k ← 1;
            end if
        end for
    end for
end for
return the best individual b
```

The search step size is the number of times that the search operator is executed. When no better solution can be found in this local neighborhood, the search step is expanded to search in a larger range. If there is still no better solution, the neighborhood structure operator is replaced.

3.4. Memetic Algorithm

The memetic algorithm [16] is a hybrid metaheuristic algorithm based on population search and local search. In this paper, the GA is used for the population search and the VNS (the IVNS without iteration) is used for the local search. In the process of population iteration, the optimal individual of each generation population is found, and the search is expanded near the optimal individual which will be replaced when a better individual is found. The procedure is as follows.

MA procedure

1. Population initialization:
   Randomly generate and evaluate every individual.
2. Iteration:
```
for i from 1 to i_max do
    GA: selection, crossover, mutation;
    Evaluation: find the best individual b;
    VNS: Take b as the initial solution to search;
    for k from 1 to k_max do
        for l from 1 to l_max do
            if find a better individual b' then
                b ← b', k ← 1;
            end if
        end for
    end for
end for
return the best individual b
```

4. COMPUTATIONAL EXPERIMENTS

To verify the effectiveness of the above algorithms, a set of examples (MK01-MK10) proposed by Brandimarte [17] are used. The distances between the stations in the job shop are random numbers between 1 and 3. The full battery capacity of AGV is set to 100. The unit no-load power consumption rate, the unit loaded power consumption rate and the unit charging rate are 1, 3 and 10, respectively. In the algorithms based on population, the population size is 60, and the crossover rate and the mutation rate are 0.95 and 0.05, respectively.

In the experiments, *Eclipse* is used as the development tool and the computer is configured with Intel (R) core (TM) i5-7200U CPU @ 2.50GHz, 2.71GHz. Taking the makespan as the evaluation standard, each experiment is carried out 20 times and the termination condition is set to evaluate the solution 100000 times. The experimental results are shown in Table 1.

Table 1 Results of the Computational Experiments

Instance	J/M/V/O	GA	IVNS	MA
MK01	10/6/3/55	80.9/75.1	81.7/75.0	**80.0**/**71.0**
MK02	10/6/3/58	**71.2**/64.0	72.4/**61.0**	72.7/63.0
MK03	15/8/6/150	267.2/248.5	260.9/235.0	**255.1**/**231.0**
MK04	15/8/4/90	113.8/107.0	112.1/102.2	**108.5**/**100.0**
MK05	15/4/4/106	**205.0**/198.0	207.2/195.0	205.3/**191.0**
MK06	10/10/6/150	180.2/164.8	**166.7**/144.0	168.1/**139.0**
MK07	20/5/4/100	**198.2**/187.0	201.6/186.0	198.5/**178.0**
MK08	20/10/9/225	563.0/532.0	560.6/527.0	**550.3**/**520.0**
MK09	20/10/9/240	448.2/442	446.2/**400.0**	**429.3**/411.0
MK10	20/15/9/240	393.3/365.0	375.5/331.0	**363.2**/**319.0**

Notes: J/M/V/O - the number of the jobs / machines / AGVs / operations; xx.x/xx.x-the average makespan/the minimum makespan; bold - the best value.

From Table 1, the following conclusions can be drawn. Firstly, when the scale of the example or the number of machines is relatively small, the difference between the three algorithms is very small and the effectiveness of the GA is relatively good. Because it is easy to find the optimal solution in a small solution space. Secondly, all the optimal solutions are obtained in the IVNS and the MA, and both have local search operators, which reflects the better ability of local search to find the optimal solution. Thirdly, with the increase of the scale of the examples, the advantage of the MA is more obvious. On the whole, the MA with both population and individual search capabilities is better. Compared with the IVNS, the MA can find a better individual among the population firstly and expand the local search in the neighborhood of the better individual, which reduces the blindness of local search and speeds up the convergence.

Taking the MK10 as an example, the convergence curves of the algorithms are shown in Fig. 3. It can be seen that the convergence speed of the MA is the fastest.

To prove the effectiveness of the dynamic scheduling strategy, machine breakdowns, AGV breakdowns and order insertions are taken as examples. The Gantt chart of initial scheduling is shown in Fig. 4.

Machine breakdowns: Machine 2 which was processing Job 1 broke down at 12:00 and put into use again after 10 minutes for maintenance, so Job 1 was scrapped and a new one was needed to be processed. The rescheduling Gantt chart of the machine breakdowns is shown in Fig. 5. The unfilled part of O_{11} in the figure indicates the processing interruption.

AGV breakdowns: AGV 3 which was transporting Job 2 broke down at 15:00 and put into use again after 8 minutes for maintenance. The rescheduling Gantt chart is shown in Fig. 6. The unfilled part of T_{23} in the figure indicates the interruption.

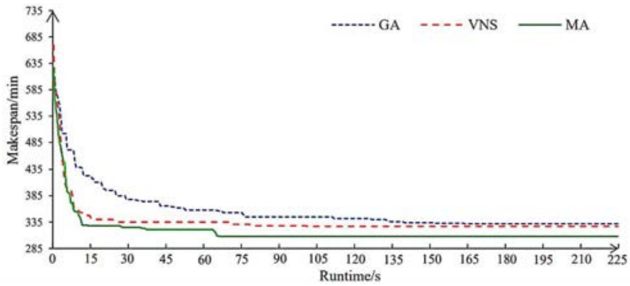
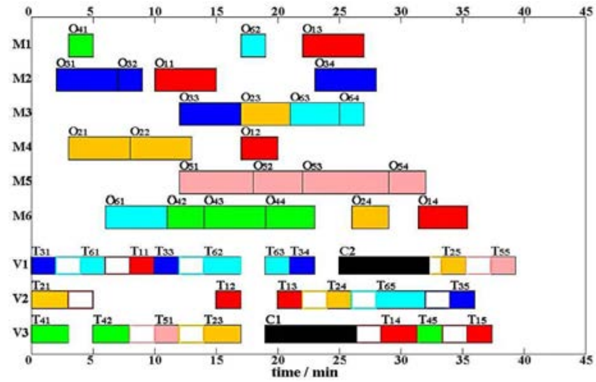


Fig. 3 Convergence Curves



Notes: M-machine; V-AGV; O_{x1} -the first operation of job x ; C-charging pile; hollow rectangle-no load; fill rectangle-process or load; T_{x5} -the virtual process of transportation back to the warehouse of job x, similarly hereinafter.

Fig. 4 Initial Scheduling Gantt Chart

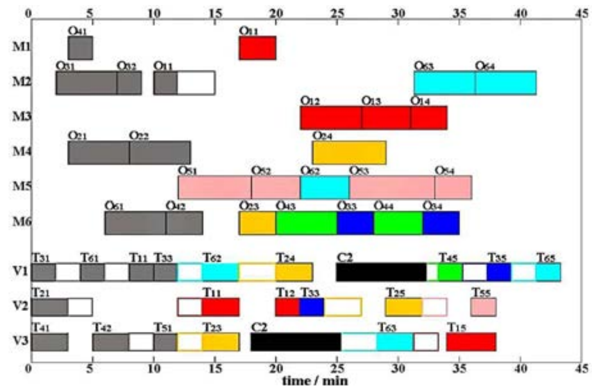


Fig. 5 Rescheduling Gantt Chart of Machine Breakdowns

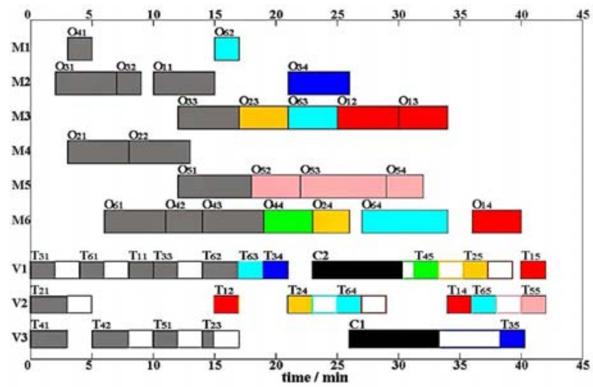


Fig. 6 Rescheduling Gantt Chart of AGV Breakdowns

Order insertions: Job 7 was inserted at 10:00. The Gantt chart of rescheduling is shown in Fig. 7.

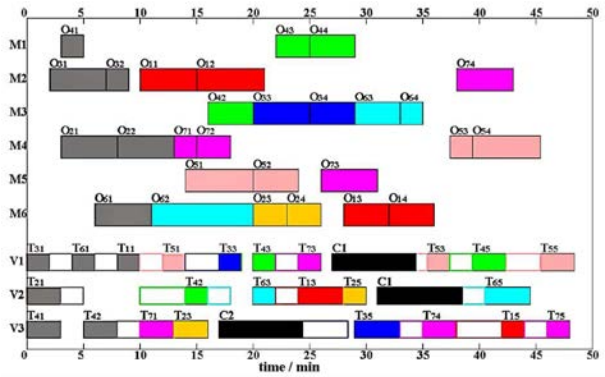


Fig. 7 Rescheduling Gantt Chart of Order Insertions

The examples of MK01-MK10 are tested with machine breakdowns and AGV breakdowns. In each experiment, the occurrence time of breakdowns is at 1/4 of initial makespan and only one machine or one AGV breaks down with 20 and 10 minutes for maintenance, respectively. The initial scheduling results are obtained after running for 3 minutes and the rescheduling running time is set to 5 seconds. Each example carried out 20 experiments and the results are shown in Table 2.

Table 2 Results of the Dynamic Computational Experiments

Instance	J/M/V/O	AM/AV	RJV	MP/AP
MK01	10/6/3/55	9.2/18.3	2.0	11.6/7.5
MK02	10/6/3/58	9.7/19.3	2.0	12.4/8.2
MK03	15/8/6/150	18.8/25.0	1.3	11.6/6.1
MK04	15/8/4/90	11.3/22.5	2.0	10.6/7.5
MK05	15/4/4/106	26.5/26.5	1.0	9.4/6.7
MK06	10/10/6/150	15.0/25.0	1.7	11.2/10.9
MK07	20/5/4/100	20.0/25.0	1.3	6.6/4.2
MK08	20/10/9/225	22.5/25.0	1.1	4.2/2.1
MK09	20/10/9/240	24.0/26.7	1.1	5.9/5.7
MK10	20/15/9/240	16.0/26.7	1.7	9.2/8.3

Notes: J/M/V/O - the number of the jobs / machines / AGVs / operations;
AM/AV - the average number of processes per machines / per AGVs;
RJV - the ratio between the number of machines and the number of AGVs;
MP/AP - the average delay time percentage of machine / AGV breakdowns;

From Table 2, for machine breakdowns, in the case of the smaller number of the machines, the smaller RJV and the larger AM, the MP is low. The larger AM indicates that the waiting time of the processes on the machines is longer. The short board of the production system lies in the processing, so the machine breakdowns will cause a larger delay. For AGV breakdowns, when the AM is smaller and the RJV is larger, the AP is high. The smaller AM means that the waiting time of the processes on the machines is shorter and the working intensity of AGV is higher. The short board of the production system lies in the transportation process, so AGV breakdowns will cause a larger delay.

In the case of short rescheduling running time (5s) that can be ignored, the percentages of delay time are within an acceptable range, so it can be seen that the algorithms and the rescheduling strategy are effective for the goal of the shortest makespan.

5. Conclusion

This paper designs three algorithms for the dynamic integrated flexible job shop scheduling problem considering material transportation. Firstly, the model of the problem with minimizing the makespan is established. Secondly, adopting the event-driven global rescheduling strategy, the GA, the IVNS and the MA are designed for the DIFJSP. Finally, the effectiveness of the algorithms and the dynamic scheduling strategy is verified by several groups of experiments.

In the future, the multi-objective scheduling and the AGV conflict-free routing will be investigated.

Acknowledgements

This work was supported by the National Outstanding Youth Talents Support Program under Grant 61822304.

References:

[1] W. H. Tan, X. F. Yuan, J. L. Wang, et al. "A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced NSGA-II: An application from casting workshop," Comput Ind Eng, vol. 160(1), July 2021.

[2] S. M. Homayouni, D. B. M. M. Fontes, "Production and transport scheduling in flexible job shop manufacturing systems," J Glob Optim, vol. 79(2), pp. 463-502, February 2021.

[3] M. Li, D. M. Lei, "An imperialist competitive algorithm with feedback for energy-efficient flexible job shop scheduling with transportation and sequence-dependent setup times," Eng Appl Artif Intell, vol. 103, August 2021.

[4] J. Yan, Z. F. Liu, C. X. Zhang, et al. "Research on flexible job shop scheduling under finite transportation conditions for digital twin workshop," Robot Comput Integr Manuf, vol. 72, December 2021.

[5] H. Wang, B. Y. Sheng, Q. B. Lu, et al. "A novel multi-objective optimization algorithm for the integrated scheduling of flexible job shops considering preventive maintenance activities and transportation processes," Soft Comput, vol. 25(4), pp. 2863-2889, February 2021.

[6] S. Luo, L. X. Zhang, Y. S. Fan, "Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning," Comput Ind Eng, vol. 159, September 2021.

[7] B. Z. Xu, Y. Mei, Y. Wang, et al. "Genetic programming with delayed routing for multi-objective dynamic flexible job shop scheduling," Evol Comput, vol. 29(1), pp. 75-105, March 2021 .

[8] H. K. Zhang, B. Buchmeister, X. Y. Li, et al, "Advanced metaheuristic method for decision-making in a dynamic job shop scheduling environment," Mathematics, vol. 9(8), April 2021.

[9] B. Luo, S. B. Wang, B. Yang, et al, "An improved deep reinforcement learning approach for the dynamic job shop scheduling problem with random job arrivals," J Phys Conf Ser, vol. 1848(1), 2021.

[10] H. Fan, H. Xiong, M. Goh, "Genetic programming-based hyper-heuristic approach for solving dynamic job shop scheduling problem with extended technical precedence constraints," Comput Oper Res, vol. 134, October 2021.

[11] I. A. Chaudhry, S. Mahmood, M.Shami, "Simultaneous scheduling of machines and automated guided vehicles in flexible manufacturing systems using genetic algorithms," J CENT SOUTH UNIV T, vol. 18(5), pp. 1473-1486, October 2011.

[12] X. Y.Li, L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," Int J Prod Econ, vol. 174, pp. 93-110, April 2016.

[13] L. Davis, "Handbook of genetic algorithm," Van Nostrand Reinhold, New York, pp. 332-349, 1991.

[14] P. Hansen, N. Mladenovic, J. A. M. Pérez, et al. "Variable neighbourhood search," COMPUT OPER RES, vol. 191(3), pp. 593-595, 2008.

[15] M. Amiri, M. Zandieh, M. Yazdani, et al. "A variable neighbourhood search algorithm for the flexible job shop scheduling problem," INT J PROD RES, vol. 48(19), pp. 5671-5689, 2010.

[16] J. H. Yang, L. Sun, H. P. Lee, et al. "Clonal selection based memetic algorithm for job shop scheduling problems," J Bionic Eng, vol. 5(2), pp. 111-119, June 2008.

[17] P. Brandimarte, " Routing and scheduling in a flexible job shop by tabu search," Ann. Oper. Res. vol. 41(3), pp. 157-183, 1993.