IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 May 2023, accepted 25 June 2023, date of publication 7 July 2023, date of current version 17 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3293029

RESEARCH ARTICLE

Research on the Integration and Scheduling of AGVs and Machines in Sample Testing Laboratory

HONGTAO TANG[1], JIAWEI WANG[1], AND ZHIXIONG YANG[2]

[1]College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China
[2]Jiaxing Sudoku Bridge Technology Company Ltd., Jiaxing 314599, China

Corresponding author: Hongtao Tang (tanght@zjut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1308102.

ABSTRACT In this paper, the problem of integration and scheduling of automated guided vehicles (AGVs) and inspection machines in medical sample-testing laboratories is addressed with the objective of minimizing the maximum turnaround time (TAT). A genetic algorithm (GA) with a sliding time window (STW) heuristic algorithm (GA-STW) is proposed for the studied problem while incorporating the vehicle assignment algorithm (VAA) for decoding operations. A mathematical model for AGV and inspection machine integration scheduling is established, and a dual-coding method for the inspection process and machine is used to solve the model. VAA was used to assign suitable AGVs to each transportation task during the decoding stage. The STW heuristic algorithm was used to further optimize the initial solution and allocate samples to appropriate inspection machines while scheduling the shortest transportation time for AGVs. The crossover operators, mutation operators, and elite numbers were experimentally optimized. A numerical simulation of standard examples demonstrated the superiority of the proposed algorithm.

INDEX TERMS Genetic algorithm, integrated scheduling, sample testing laboratory, sliding time window.

I. INTRODUCTION

With the continuous development of information, logistics, and automation technologies and their widespread application in the healthcare industry, total laboratory automation (TLA) has become the goal of medical sample-testing laboratories [1]. This technology connects sample transportation systems, transport robots, automatic analyzers, process control, and related software and hardware [2], enabling the automatic processing and tracking of samples, reducing the TAT of test samples, improving laboratory productivity, and reducing costs. The TAT refers to the duration from the arrival of a test sample at the laboratory to the release of the test results. This is an essential component of comprehensive quality management in the laboratory, reflecting the timeliness of generating test reports for sample analysis. The TAT serves as the standard for measuring efficiency. In this system, the key to improving the TAT and testing efficiency is the joint scheduling and optimization of sample transportation, AGV task allocation and scheduling, and testing equipment scheduling. It is essentially a joint scheduling optimization problem for samples, AGVs, and testing equipment.

The sample testing laboratory is composed of various biomedical instruments and testing machines. Depending on the specific testing requirements, the corresponding reagent type and testing machines need to be considered for different testing projects to conduct appropriate testing operations on different samples. Automation can enhance the efficiency of a laboratory and reduce safety risks, as demonstrated by studies conducted by Lam and Jacob [3], Seaberg et al. [4], Sarkozi et al. [5], and Melanson et al. [6]. Scholars from different countries have conducted comprehensive research on reducing TAT in sample testing laboratories, which can be categorized into strategic decision-making [7], [8], tactical decision-making [9], [10], [11], and management decision-making [12], [13], [14], [15]. However, there has been limited research on the integration and scheduling of AGVs and machines in sample testing laboratories. Therefore, this study

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Ching Ying.

primarily focuses on the integration and scheduling problem of AGVs and machines with the goal of minimizing maximum TAT in the laboratory.

The integrated scheduling problems of AGVs and machines primarily involve machine allocation, process sequencing, AGV allocation with task handling, and AGV path planning [16]. The abstraction and simplification of an actual problem are often required during modeling. The constraint conditions of models in the literature can be classified into three categories: process, machine, and AGV. Some scholars have considered special constraints when modeling, such as the loading and unloading times studied by He et al. [17]; however, they were incorporated into the machine processing time for handling. Lyu et al. [18], He et al. [17], Zou et al. [19], and Deng et al. [20] achieved integrated scheduling under the premise of no-conflict paths by introducing time windows into the model. Li et al. [21] and Li and Liu [22] considered the impact of the charging factors on scheduling in their research. In the static scheduling problem of AGVs, some scholars have also considered the multi-load environment of AGVs, including Liu et al. [23], Ge et al. [24], and Wu et al. [25], which helps optimize the AGV travel route and reduce the running distance. However, there is still little research on factors such as workpiece arrival time and delivery time. Ma et al. [26] considered delivery time constraints and minimized the total delay as one of the optimization objectives.

Optimization algorithms for studying such problems can generally be classified into four categories: exact algorithms, heuristic algorithms, intelligent optimization algorithms, and simulation methods. Owing to the complexity of the AGV and machine integration scheduling problems, current optimization algorithms mainly focus on the latter two categories. In recent years, scholars have improved these algorithms to obtain optimal solutions. For example, Homayouni and Fontes [27] proposed a biased random key genetic algorithm based on operations with multiple starts. Zou et al. [28] presented a hybrid genetic algorithm based on time windows and the Dijkstra algorithm. Yuan et al. [29] introduced a hybrid genetic algorithm based on simulated annealing. Dai et al. [30] developed an improved genetic algorithm to minimize the maximum completion time, AGV transportation energy consumption, machine processing energy consumption, and other resource energy consumption. Li and Lei [31] proposed a feedback-based imperialist competitive algorithm to minimize the above-mentioned four objectives and average job delay time, achieving satisfactory results. Furthermore, Qin et al. [32] addressed sequence ordering, factory allocation, machine selection, and job blocking using a collaborative iterative greedy (CIG) algorithm. Similarly, an improved exchange based iterative greedy search (IGS) algorithm was proposed in [33] to solve this problem. Two perturbation strategies were designed to enhance the local and global search capabilities of this solution and reduce the impact of blocking constraints on job sequences. The paper [34] introduces a new flow shop combinatorial optimization problem was introduced, and an iterative greedy algorithm that includes two key technologies was proposed. One is the decoding process for calculating the completion time of job sequences, and the other is the neighborhood probability selection strategy for jobs with families and blocks. Additionally, emerging hybrid algorithms, such as whale optimization based on genetic algorithms [35], multi-objective evolutionary algorithms based on reverse learning strategy [36], improved flower pollination algorithm [37], and hill-climbing algorithm based on delayed acceptance strategy [27], have also been introduced.

Based on the above analysis, current research in the context of medical sample testing laboratories generally focuses on improving TLA devices, training personnel, and machine upgrades. However, with the gradual widespread application of AGVs in hospitals, research on the integration and scheduling issues of AGVs and machines in this context is limited. Furthermore, most of the proposed algorithms are not applicable to the problem studied herein, and the research models related to integrated scheduling problems are mostly based on standard cases. Therefore, this article addresses the integrated scheduling problem in the context of a sample testing laboratory, taking into account the peculiarities of medical testing, such as the sample arrival time, limitations in testing projects that can be completed by each machine, AGV scheduling, and latest testing time, which must be considered to minimize the TAT as the objective of establishing a model that embeds the AGV and machine scheduling subproblems in the iterative solution process. To avoid premature convergence and the occurrence of local optima, a GA-STW algorithm was developed to address the integration and scheduling problem between the inspection machine and AGVs in the sample testing laboratory. The parameters that fit the model proposed in this study were calculated. The algorithm first uses a GA to encode the processes and testing machines separately, and then generates machine scheduling solutions. During decoding, a VAA is introduced to optimize the AGV scheduling. To prevent blank time windows in feasible solutions, a STW heuristic algorithm seeks better solutions given machine scheduling to minimize the maximum TAT and improve the efficiency of the laboratory.

II. PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

A. DESCRIPTION OF SAMPLE TESTING LABORATORY INTEGRATED SCHEDULING PROBLEM

This study investigates the integrated scheduling of AGVs and machines in sample-testing laboratories, with the goal of minimizing the TAT of the laboratory. The problem considers the scheduling of both testing machines and AGVs as well as constraints such as the arrival time of test samples, processing (testing) time, transport time, and deadline time. A dual-constraint model for transporting AGVs and testing machines was established. TAT refers to the time from when a doctor requests a test project to when the patient obtains the test report and includes nine steps: doctor requests, specimen collection, confirmation, transport, preprocessing, testing, result reporting, analysis, and corresponding measures. This study mainly considered a narrow definition of TAT: the time from transportation of the sample to the laboratory to the production of the result, also known as laboratory TAT [38]. Test samples were placed in batches on a sample rack and delivered to the laboratory. The sample racks were placed on the table, with each sample rack containing r test samples; inspection there were n sample racks $\{J_1, J_2, \ldots, J_n\}$. The samples were first sent to the preprocessing machine and then transported to the corresponding testing machine {s$\{M_1, M_2, \ldots, M_m\}$ for testing. After testing was completed, the samples were placed in the buffer area of the testing equipment and transported to the refrigerator for storage. The sample transfer between each link in the testing process was completed using v identical AGVs $\{R_1, R_2, \ldots, R_l\}$. The sample preparation process is illustrated in Fig. 1.

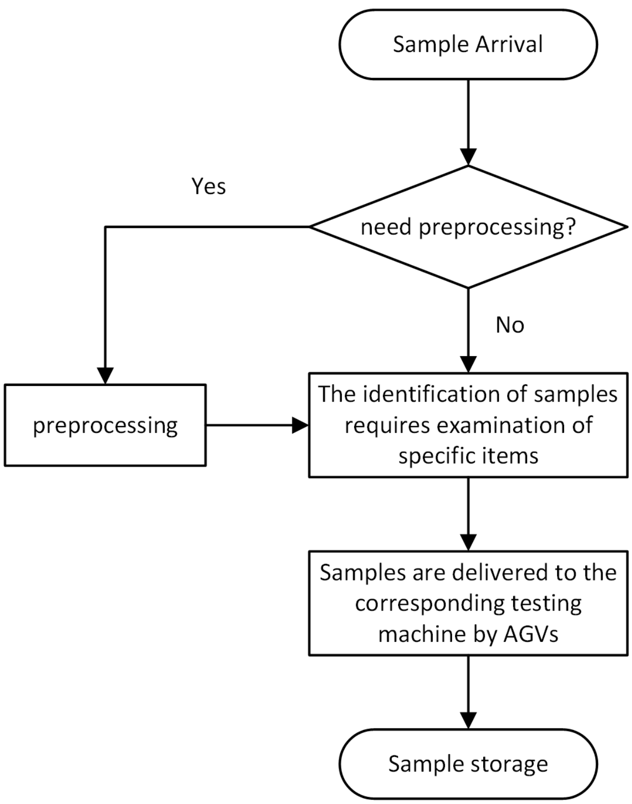


FIGURE 1. **Sample flow chart.**

Owing to inconsistent reagent positions in the different testing machines, the available test items were not completely consistent. In this study, it is necessary to select suitable testing equipment for each testing process and appropriate AGVs for the transportation task to ensure that the samples are tested within a specified time and to improve the testing efficiency.

To better analyze the problem, the following assumptions are made:

1) Information such as the selected testing machine, processing time, arrival time, and deadline is known for each testing sample.
2) At time 0, all AGVs and testing machines are in an idle state.
3) The testing machines have infinite buffer areas.
4) The testing samples in each sample rack have no priority.
5) The testing equipment can only test one sample rack at a time.
6) Machine failure, reagent replacement, and other factors are not considered.
7) The reagent positions and testable items are not identical.
8) The testing sample cannot be stopped once the testing process begins.
9) An AGV can transport only one sample rack at a time, and the transportation process cannot be interrupted.
10) AGV failures, charging, congestion, and other conditions are not considered; the AGVs operate at a constant speed.

B. MATHEMATICAL MODEL

1) PARAMETER SYMBOLS AND DEFINITIONS

J_i: set of sample racks, J $= \{J_1, J_2, \ldots, J_n\}$, n = number of sample racks;

M_k: set of machines, $M = \{M_1, M_2, \ldots, M_m\}$, m = number of testing machines;

R_v: set of AGVs, $R = \{R_1, R_2, \ldots, R_l\}$, l = number of AGVs;

O_{ij}: indicates the j-th testing process of $\mathrm{J_i}$, $j = \{1, 2, \ldots, \mathrm{h}\}$;

G_i: indicates the number of test samples in the i-th sample rack;

t_k: indicates the time for the k-th testing machine to test the sample;

S_{ijk}: indicates the start time of O_{ij} on M_k;

E_{ijk}: indicates the end time of O_{ij} on M_k;

T_i: indicates the TAT of sample rack i;

A_i: indicates the arrival time of sample rack i;

D_i: indicates the deadline of sample rack i;

P_{ijk}: indicates the testing time of O_{ij} on M_k;

$O_{i'j'}$: indicates the preceding testing process of O_{ij} on the current testing device;

$S_{i'j'k'}$, $E_{i'j'k'}$, $P_{i'j'k'}$: indicates the start time, end time, and testing time of $O_{i'j'}$ on M_k;

T_{vij}: indicates the time required for AGV to transport O_{ij} for delivery;

T_{bv}, T_{ev}: indicates the start and end time for AGV to be empty;

T_{bvij}, T_{evij}: indicates the start and end time for AGV to transport O_{ij} for delivery;

MS_k: indicates the start time of M_k being idle;

ME_k: indicates the end time of M_k being idle;

M_{vij}, M'_{vij}: indicates the machine where sample rack i is located and the machine where AGV is located during the delivery of O_{ij};

L: indicates a sufficiently large positive number

Decision variables:

$$X_{ijk} = \begin{cases} 1, & \text{if } O_{ij} \text{is tested on } \mathrm{M_k} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$Y_{ijv} = \begin{cases} 1, & \text{if } O_{ij} \text{is transported by} R_v \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\mathrm{X}P_{pqijk} = \begin{cases} 1, & \text{if } O_{pq} \textit{ is tested on } M_k \textit{ prior to } O_{ij} \textit{ testing} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$\mathrm{Y}P_{pqijk} = \begin{cases} 1, & \text{if } O_{pq} \textit{ is transported earlier than } O_{ij} \textit{ on } M_k \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\mathrm{Z}\left(\mathrm{M_k}\right) = \begin{cases} 1, & \text{if The current testing machine is in use} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

2) MODEL REPRESENTATION

$$\mathrm{f} = \min\left(\max(T_1, T_2, \ldots, T_i)\right) \tag{6}$$

$$\sum_{\mathrm{k}=1}^{\mathrm{m}} X_{ijk} = 1 \tag{7}$$

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} X_{ijk} = 1 \tag{8}$$

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} Y_{ijv} = 1 \tag{9}$$

$$S_{ijk} + P_{ijk} + T_{vi(j+1)} \le S_{i(j+1)k} \tag{10}$$

$$S_{i'j'k} + P_{i'j'k} \le S_{ijk} \tag{11}$$

$$E_{ijk} = S_{ijk} + P_{ijk} \tag{12}$$

$$S_{ijk} + \mathrm{L}(1 - \mathrm{X}P_{pqijk}) \ge E_{pqk} \tag{13}$$

$$T_{bvij} + \mathrm{L}(1 - \mathrm{Y}P_{pqijk}) \ge T_{evpq} \tag{14}$$

$$T_{bvij} \ge E_{ijk} \tag{15}$$

$$T_i \le D_i \tag{16}$$

$$P_{ijk} = X_{ijk} * (G_i * t_k) \tag{17}$$

$$MS_k \ge E_{i'j'k}, ME_k \ge S_{ijk} + P_{ijk} \tag{18}$$

$$T_{vij} = \mathrm{T}\left(M'_{vij}, M_{vij}\right) + \mathrm{T}\left(M_{vij}, M_k\right) \tag{19}$$

$$X_{ijk} = 0,\ 1 \le \mathrm{i} \le \mathrm{n},\ \mathrm{k} \notin M_i \tag{20}$$

$$T_{evij} = T_{bvij} + T_{vij} \tag{21}$$

$$T_i = E_{ihk} \tag{22}$$

The goal of the objective function (6) is to minimizes the maximum TAT. Constraint (7) indicates that the sample rack can be inspected using only one testing machine at a time. Constraint (8) indicates that the machine can inspect only one sample rack at a time. Constraint (9) indicates that an AGV can transport only one sample rack at a time. Constraint (10) indicates the constraint between two adjacent inspection processes for the same sample. Constraint (11) indicates the inspection constraint of the previous process for the same inspection device. Constraint (12) indicates the completion time for the inspection O_{ij}. Constraint (13) indicates the inspection capacity limitation of the testing machine. Constraint (14) indicates the capacity limitation of the AGV. Constraint (15) indicates that the AGV must transport the sample rack after inspection is completed. Constraint (16) represents the deadline constraint for an inspected sample. Constraint (17) represents the inspection time of the sample rack i on machine k. Constraint (18) indicates that a sample can only be inspected when the machine is available. Constraint (19) represents the transportation time of O_{ij} by the AGV. Constraint (20) indicates that the sample rack can only be scheduled for machines that inspect the type of sample. Constraint (21) indicates that the completion time of the transportation task by the AGV is the sum of the start time of transporting the sample rack and transportation time. Constraint (22) represents the TAT of the sample rack i, which is the completion time of the last inspection process for that sample.

With the specificity of sample testing in the laboratory, this study also considers whether samples need to undergo pretreatment to creat constraints. Depending on the availability of the testing machine, the following conditions are considered.

If there is a pretreatment process for the sample and the testing machine is available, the start time of the test is the sum of the sample rack arrival time and the time required for the AGV to transport the sample rack. If the testing machine is unavailable, the start time of the sample test is determined as the maximum time between the arrival time, time required for the AGV to transport the sample rack, and idle time of the machine.

$$S_{i1k} = \begin{cases} A_i + T_{vij}, & \mathrm{Z}\left(\mathrm{M_k}\right) = 0 \\ \max\left(A_i + T_{vij}, MS_k\right), & \mathrm{Z}\left(\mathrm{M_k}\right) = 1 \end{cases} \tag{23}$$

III. GA-STW ALGORITHM DESIGN

A GA-STW algorithm is proposed for the model in this study. Machine scheduling and AGV scheduling problems are embedded in the iterative solving process. The main idea of the algorithm is to apply the MSOS-I encoding method [39]. During the decoding process, the VAA is used to process the AGVs. The VAA attempts to search for the AGV with the shortest transportation completion time in each transportation route to minimize the transportation time and start-time deviation for each operation. After generating the initial scheduling results, a STW heuristic algorithm [40] is used to further optimize the solution to the problem. STW checks whether a given number of AGVs can complete all transportation routes within the time window generated by the machine scheduling subproblem. Based on the different conditions, the time window was adjusted while considering the idle status of the AGVs and the earliest start time for sample inspection to construct a new time window. This algorithm can effectively improve the inspection rate without violating any constraints, while avoiding long waiting times for sample inspection and AGV transportation.

A. ALGORITHM ENCODING AND DECODING

1) ALGORITHM ENCODING

Owing to the unique nature of the model used in this study, both AGV and inspection machine scheduling must be considered. Thus, the problem can be decomposed into two subproblems: the machine allocation problem, which selects machines that meet the constraint conditions and optimization goals for each process, and the AGV allocation problem, which arranges suitable AGVs for each process and determines the starting and ending positions of the AGVs and their corresponding time nodes for each transport until all sample racks are inspected. As the encoding and solution spaces are mapped, the encoding method affects the size of the search space for the subsequent algorithm searches. An encoding strategy based on the inspection process and machine was designed for the model constructed in this study, as shown in Figure 2. The encoding method is divided into machine selection and process order strings. The AGV allocation subproblem is solved using the VAA algorithm and completed during the decoding process.

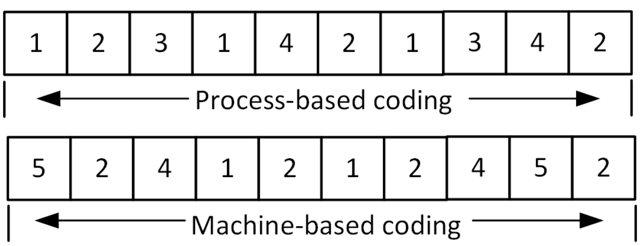


FIGURE 2. Chromosome encoding.

Each chromosome consists of two parts. The first part is encoded based on the testing process; the number appearing at the gene locus represents the sequence of the sample tray in the testing process. For example, the number 3 on the third gene locus represents the first process of sample tray 3, number 1 on the fourth gene locus represents the second process of sample tray 1, and so on.

The second part is encoded based on the testing machine, and the number at the gene locus represents the selected machine number for the corresponding testing process. For example, number 5 on the first gene locus indicates that the first process of sample tray 1 is tested using machine 5; number 2 on the second gene locus indicates that the first process of sample tray 2 is tested using machine 2, and so on.

2) ALGORITHM DECODING

The aforementioned encoding operation can solve the problem of sample rack selection in an inspection machine, and the AGV scheduling is implemented in the decoding process. In this study, a VAA is used for processing, which considers the scheduled inspection process name, start time, and name of the previous the inspection process as the input information received from the chromosome decoding process. It is a greedy algorithm that can achieve good results in a short time after determining the arrangement of inspection processes and inspection machines. It searches for the best AGV to obtain the earliest inspection start time on the designated inspection machine.

TABLE 1. Sample test data.

Sample	Operation	Machine	Processing time
1	1	M1	10
2	1	M2	12
3	1	M3	7

TABLE 2. Transport time matrix.

From/To	Loaded	M1	M2
Loaded	0	3	10
M1	3	0	5
M2	10	5	0

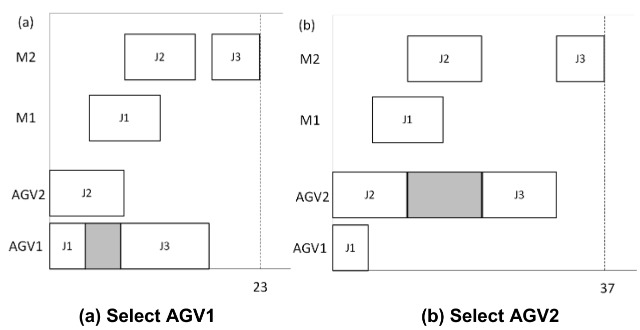


FIGURE 3. An illustration of the VAA.

Table 1 and Table 2 present the data used to illustrate the decoding methods. This paper assumes that the VAA considers how to transport J_3 after arranging the first operations of J_1 and J_2. At this point, the first operation of sample rack 3, from the inspection table to the corresponding first inspection equipment, was assumed to be inspection machine M_2. Two scenarios were considered, as shown in Figure 3.

1) Figure 3.(a) shows that AGV1 is used for transportation. AGV1 is located on machine M1. When dispatching this AGV, AGV1 travels empty from M1 to the inspection station, and then loads J_3 from the

inspection station to machine M_2. The inspection of rack 3 was completed at 23.

2) Figure 3.(b) shows that AGV2 is used for transportation. At this time, AGV2 is located on machine M_2. After dispatching AGV2, it travels empty from M_2 to the inspection station, loads J_3 to the inspection station, and returns to M_2. The inspection of rack 3 was completed at 37.

From Figure 3, it can be observed that selecting AGV1 for transportation in this case results in a shorter completion time for the inspection of sample rack 3. Thus, AGV1 was chosen to handle the transportation task for sample rack 3. The main decoding concept of VAA is to stretch the generated machine schedule when considering the transportation time of AGVs for a given operation sequence represented by a process-based encoding method. This is because the start time of each process changes. In other words, ideally minimizing the start time of the processes can generate the shortest overall time for a combined schedule of AGVs and machines. The VAA attempts to determine the AGV with the shortest transportation time for each run to minimize the offset in the start time of each process.

B. ALGORITHMIC OPERATOR DESIGN

1) SELECTION OPERATORS

The selection operation can increase the survival probability of genes with higher fitness, ensuring the passing of excellent traits from the parent generation to their offspring, effectively improving the efficiency of the algorithm. This study refers to the operation of selecting two parent strings to generate a new string (i.e., a substring), as used by Murata et al., which has demonstrated good performance in solving such problems and can achieve stronger selection intensity [41]. Let N_{pop} be the number of solutions in each population in the genetic algorithm; N_{pop} is the population size. The notation $\Psi_t = \{x_t^1, x_t^2, \ldots, x_t^{Npop}\}$ is used to represent the N_{pop} solutions in the t-th generation. Each solution x_t^i is selected as a parent string based on its selection probability, $Ps(x_t^i)$. The following selection probabilities are used in the simulation:

$$P_s\left(x_t^i\right) = \frac{\left[f_M\left(\Psi_t\right) - f\left(x_t^i\right)\right]^2}{\sum_{x_t^i \in \Psi_t}\left[f_M\left(\Psi_t\right) - f\left(x_t^i\right)\right]^2} \tag{24}$$

Function $f(x_t^i)$ is the objective function (makespan) to be minimized in scheduling problems, and $f_M(\Psi_t)$ is the worst value of f in the t-th generation.

$$f_M\left(\Psi_t\right) = \max\{f(x_t^i) | x_t^i \in \Psi_t\} \tag{25}$$

2) CROSS OPERATOR

The crossover operator is an important component of the GA. Utilizing the advantageous genes of chromosomes and ensuring feasible solutions can help inherit better features from the fittest solutions. Based on the characteristics of the problem in this study, two different crossover methods are applied to the process and the machine.

First, we randomly divide all sample racks into two groups $\{J_1, J_2\}$, randomly select two different parental process chromosomes, reserve the process positions belonging to J_1 in P_1 to C_1, reserves the process positions belonging to J_2 in P_2 to C_2, copy the process positions belonging to J_2 in P_2 to the remaining positions in P_1, and copies the process positions belonging to J_1 in P_1 to the remaining positions in P_2. The crossover process is illustrated in Figure 4.

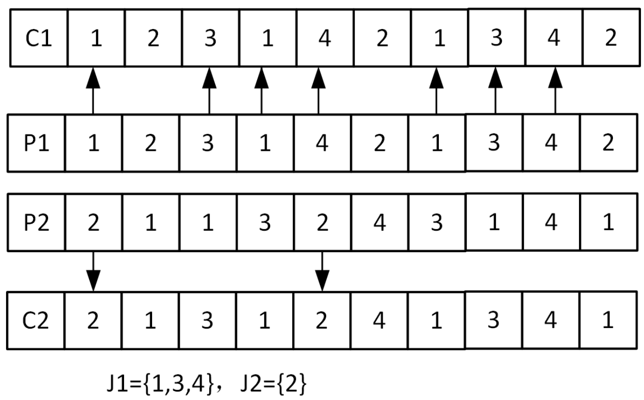


FIGURE 4. Cross-operational process of procedures.

Subsequently, a cross is performed on the testing machines. Two different parent machine chromosomes are selected, and the positions of the procedure for machine cross-validation are randomly generated as the crossing positions of parent P_1'. Cross-validation is performed at the corresponding position on P_2. For example, the first process of sample rack 3 in P_1 is selected as the crossing point, and the corresponding position is determined in P_2 for crossing. The crossing process is illustrated in Figure 5.

3) MUTATION OPERATORS

Mutation operations help the GA to reach as many points as possible in the search space. They can be viewed as searching from the current solution to the local solutions. This study uses different mutation methods for processing and inspection machines, described as follows.

First, there is a variation in the process. Using the exchange mutation method, a segment of the parent chromosome is selected, and only two positions of the process are randomly selected and exchanged. To prevent the testing machine from becoming unavailable, adjustments must be made according to the process-exchange method. For example, after exchanging the first process of sample rack 1 with the first process of sample rack 4, the corresponding testing machine was changed, and the machines needed to be exchanged simultaneously. In other words, the testing machine was changed from machine 5 to machine 2 after the exchange. The mutation process is illustrated in Figure 6.

Next, this paper focuses on the variations in the testing machine. A segment of the parental chromosome is selected,

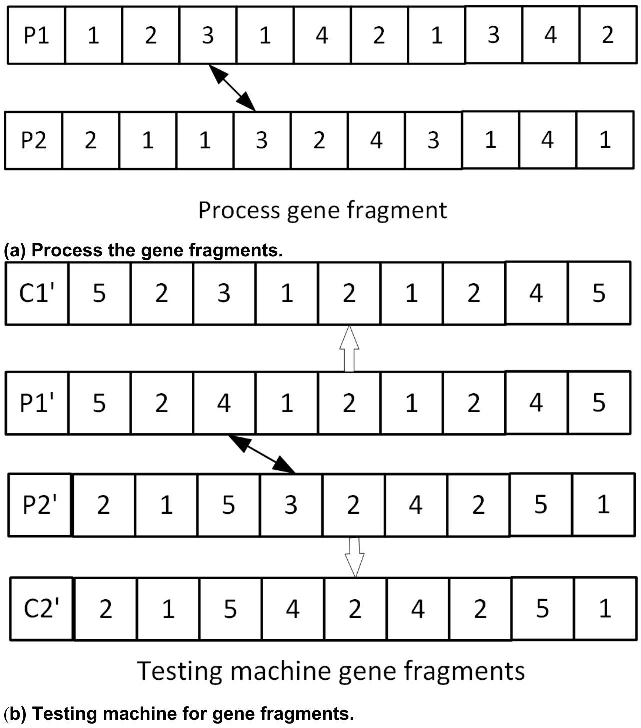


FIGURE 5. Verification of cross-operation process of machinery.

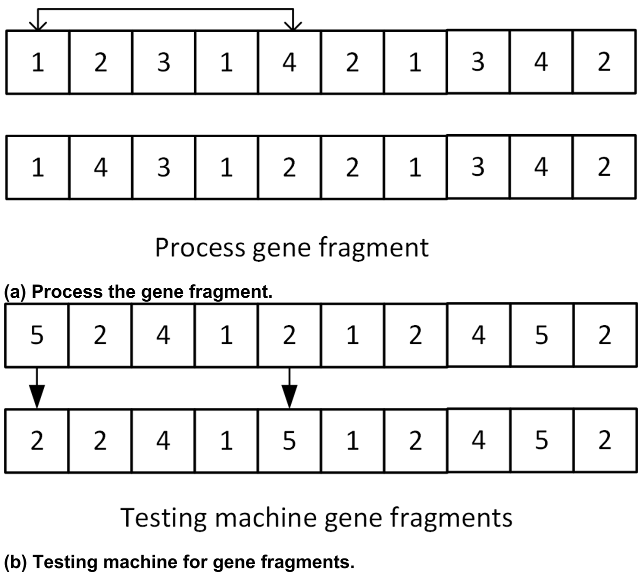


FIGURE 6. Process of variation in operation procedures.

and multiple mutation points are randomly generated. The machine is reselected from the machine set corresponding to the testing process. For example, when the first process position of sample rack 3 undergoes a mutation, the available machine set is $\{J_3, J_4\}$. A random number with a length equal to that of the set is generated, and the corresponding machine is selected based on a random number. The mutation process is illustrated in Figure 7.

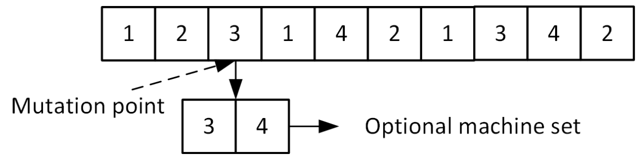


FIGURE 7. Verifying the variation of machine operation process.

4) ELITE RETENTION STRATEGY

An elite individual is the individual with the highest fitness value discovered by the genetic algo5rithm during the evolution of the population, possessing the best genetic structure and advantageous traits. The advantage of using elite reservation is that the genetic algorithm does not lose or destroy the most optimal individual that has appeared thus far during the evolutionary process in the selection, crossover, and mutation operations. The elite reservation strategy plays a significant role in improving the global convergence ability of standard genetic algorithms. Rudolph [42] theoretically demonstrated that a standard genetic algorithm with elite reservation is globally convergent.

This study uses the elite retention method proposed by De Jong [43], assuming that a(t) is the best individual in the population in the t-th generation. Let A(t+1) be the new-generation population. If there is no individual in A(t+1) better than a(t), then add a(t) to A(t+1) as the $(n+1)$-th individual, where n is the size of the population. To maintain a constant population size, if an elite individual is added to the new-generation population, the individual with the minimum fitness value can be eliminated.

C. STW HEURISTIC SCHEDULING ALGORITHM

A feasible solution to the problem model can be obtained using GA encoding and VAA decoding. However, in the solution process, the computation is based on the assumption of an infinite number of idle AGVs. In reality, the number of AGVs is limited. When introducing the AGV quantity limitation, the condition where the machine inspection completion time and the available time point for calling AGVs in the previous section overlap and whether there is a time difference between the transportation time and the machine inspection completion time must be considered. AGVs are scheduled to be dispatched only after completion of the inspection; it must also be considered whether idle AGVs are available for transportation at that time. If idle AGVs are available for transportation and can travel empty, the optimized time at this point is the time for the AGV to reach the machine from its current position. This study introduces an STW to further investigate this condition.

1) OVERVIEW OF STW ALGORITHM

An STW does not divide the fixed start and end points of time windows. Instead, it uses the arrival time of each request as the endpoint of the statistical time window. The starting point is the time point obtained by pushing back the length of the time window from the endpoint [40]. An STW divides the time window into smaller intervals. Each time an interval passes, the time window slides one grid to the right and each interval has an independent counter. When calculating the total number of requests in the entire time window, all the counters in each time interval are accumulated.

Problems that can be solved using the STW approach typically have elements within the window that are continuous, meaning that the abstract sliding window is continuous on the original array or string, and can only slide from left to right and not backward. This means that the left and right boundaries of the window can only increase from left to right and cannot decrease, even locally [44]. The algorithmic approach for a simple STW is presented as follows:

1) Using the left-right pointer technique, we initialize left = right = 0 and refer to the closed interval [left, right] as a [window].
2) The right pointer is continuously increased to expand the window [left, right] until it meets the requirements.
3) The window [left, right] is narrowed by stopping the right pointer and continuously increasing the left pointer until the string in the window no longer satisfies the requirements. The result is updated for one round with each increase in the left pointer.
4) Steps 2 and 3 are repeated until the right side reaches the end.

Step 2 is equivalent to finding a feasible solution and Step 3 determines the optimal solution. The left and right pointers alternately move forward; the window size constantly increases and decreases, causing the window to slide to the right.

2) STW ALGORITHM OPTIMIZATION STEPS

Using the STW to further optimize the model, first constrain the time window W_i.

$$C_{(i-1)jm} + T_{vij} \leq W_i \leq C_{ijk} - P_{ijk} \tag{26}$$

As shown in Figure 8, when $O_{i(j-1)m}$ is tested in the j-1st process at station M_m, the inspection completion time is $E_{i(j-1)m}$. The next inspection for sample rack i was carried out at station M_k; the completion time of $O_{i'j'}$ tested at M_k is $E_{i'j'k}$. Assuming that there is a free AGV at the time point $E_{i'j'k}$, the inspection of sample i can be performed immediately after completion of the inspection of sample rack i', which means that the earliest inspection time for sample rack i is $E_{i'j'k}$. However, according to the GA and VAA algorithms, the instruction to transport sample rack i by the AGV is triggered only when the time point $E_{i'j'k}$ is reached, resulting in the actual inspection time of sample rack i being:

$$S_{ijk} = E_{i'j'k} + T_{vij} \tag{27}$$

In an ideal scenario, when there is a blank time period, if there is a free AGV at time $E_{i(j-1)m}$, then the AGV can

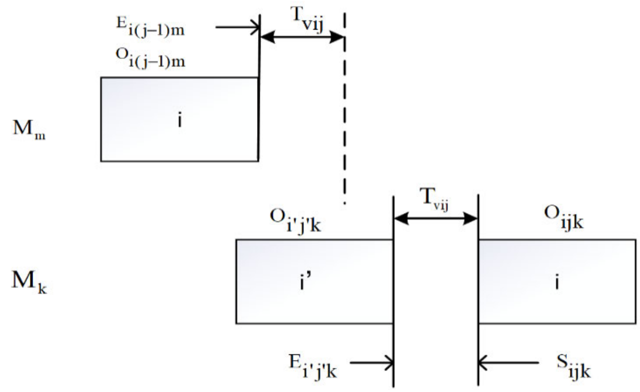


FIGURE 8. A blank time window condition.

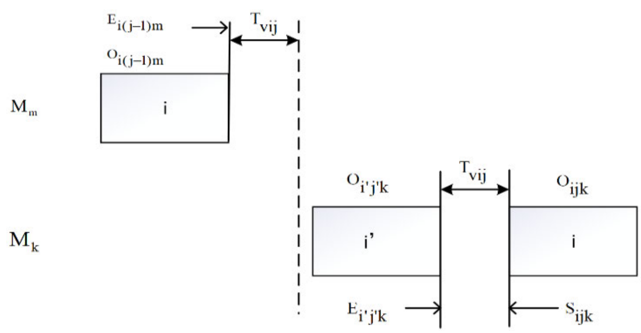


FIGURE 9. STW scenario 1.

be scheduled to transport sample rack i at that time, and it can arrive at M_k for inspection at time $E_{i'j'k}$. To avoid blank periods, this study introduces an STW algorithm to optimize the problem. To determine the optimal solution for the model, it is necessary to analyze different conditions and consider the relationship between the time point $E_{i'j'k}$ and the availability of AGVs at time $E_{i(j-1)m}$. Two conditions must be considered.

1) As shown in Figure 9, when $E_{i'j'k} \geq E_{i(j-1)m} + T_{vij}$, if there is a free AGV at time $E_{i(j-1)m}$, the AGV with the shortest transportation time is selected to transport sample rack i. The starting time of the sample rack inspection is:

$$S_{ijk} = E_{i'j'k} \tag{28}$$

If there is no available AGV at that moment, the slide time window W_i to the right and check whether there is any available AGV at time $E_{i'j'k} + W_i$. If so, select the AGV; otherwise, continue sliding the time window until an available AGV is obtained. The starting time of the sample rack inspection is presented as follows, where n is the number of STW.

$$S_{ijk} = E_{i'j'k} + n * W_i \tag{29}$$

2) As shown in Figure 10, at this time $E_{i'j'k} \leq E_{i(j-1)m} + T_{vij}$. If there is an available AGV at time $E_{i(j-1)m}$, the AGV with the shortest transport time is selected to transport sample rack i. The starting time of the sample

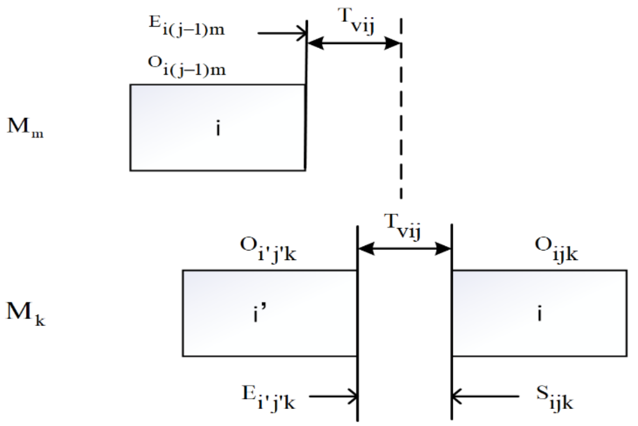


FIGURE 10. STW scenario 2.

rack inspection is:

$$S_{ijk} = E_{i(j-1)m} + T_{vij} \tag{30}$$

If no AGV is available at this time, the time window W_i slides to the right. This study examines whether there is a free AGV at $E_{i'j'k}+W_i$. If there is one, select AGV; otherwise, slide through the time window until there is an available AGV. The starting time of the sample rack is:

$$S_{ijk} = E_{i'j'k} + T_{vij} + n*W_i \tag{31}$$

The STW concept is consistent with the subproblem of AGV scheduling; given a solution to the machine scheduling subproblem, there are specific operation sequences and specific completion times C_{max} for each machine. The earliest and latest completion times are calculated by sliding the STW backward [45]. The main process framework of STW is shown in Figure 11.

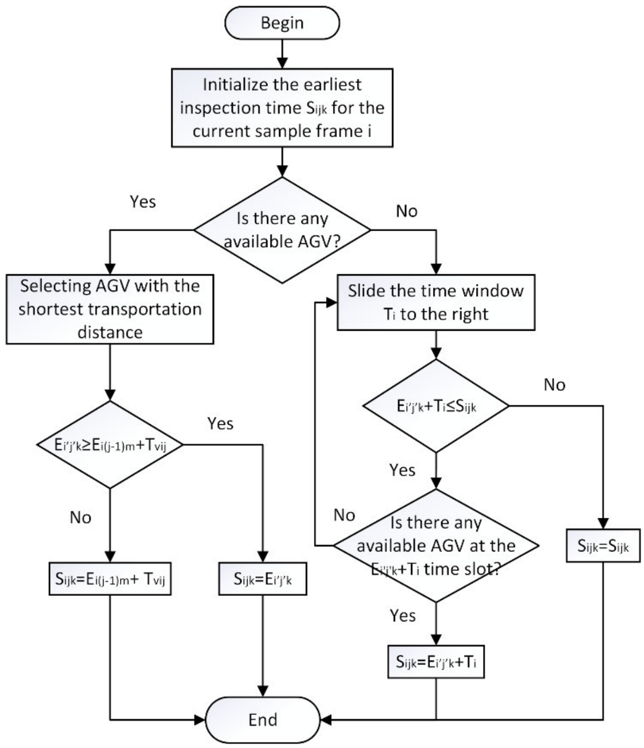


FIGURE 11. The main process framework of STW.

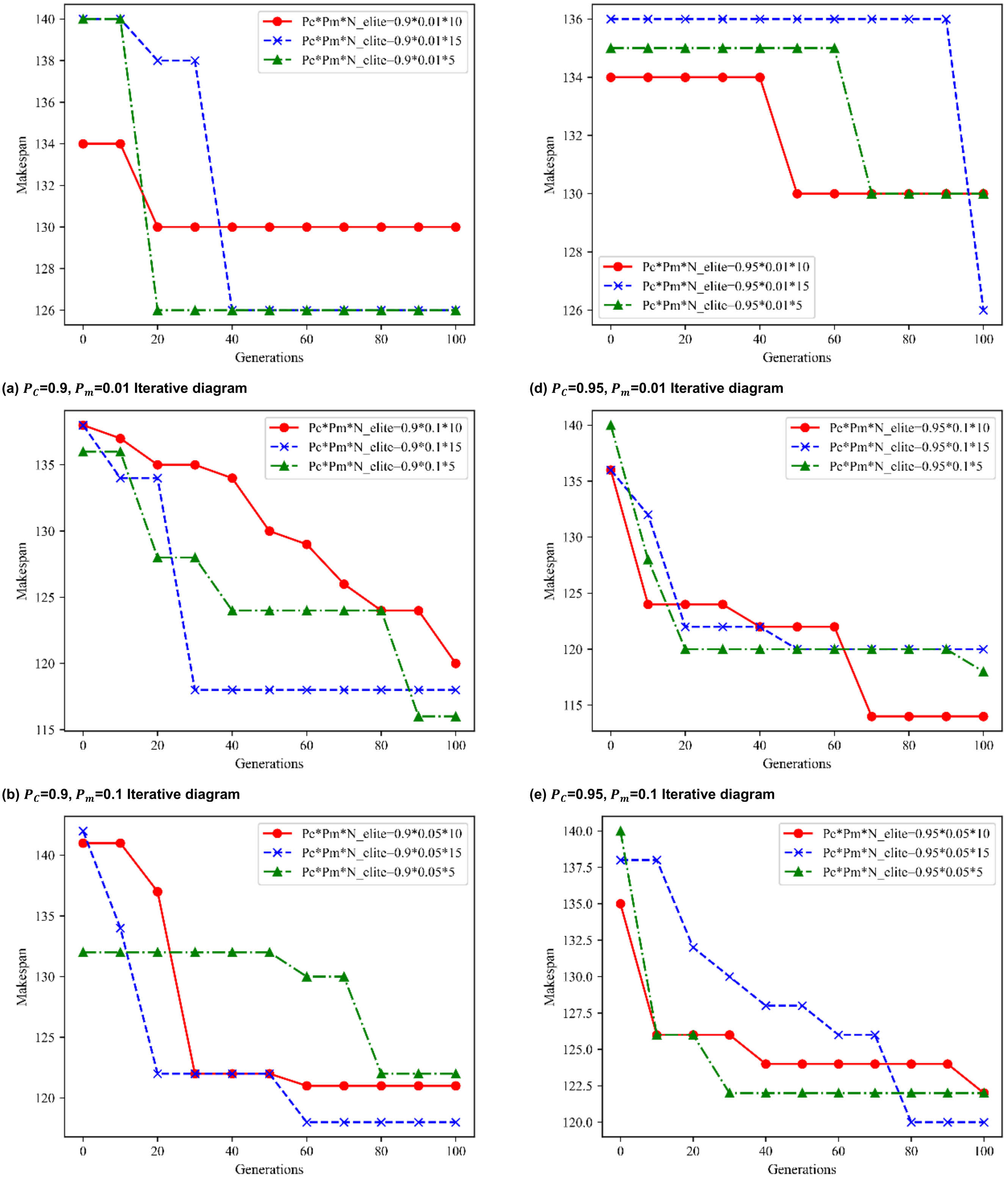
IV. EXPERIMENTAL AND COMPUTATIONAL RESULTS

To determine and verify the performance of the GA-STW algorithm proposed in this study to solve the integrated scheduling problem between AGVs and inspection machines in hospital laboratories, experiments were conducted in a Python 3.11 programming environment on a processor with Intel(R) Core(TM) i5-7300HQ CPU @ 2.50 GHz and 16.0 GB of memory. This study focuses on the integrated scheduling problem between AGVs and inspection machines in a sample-testing laboratory. Because of the specific nature of this problem, it differs significantly from typical workshop-scheduling problems. In this study, the problem size is set as n = 5 sample racks, m = 4 inspection machines, and l = 2 AGVs. Additionally, during inspection, some samples required preprocessing before testing, whereas others did not, resulting in two or three processing steps in the model. The corresponding parameters are set and the experiments are conducted.

A. COMPARATIVE ANALYSIS OF ALGORITHM PARAMETER OPTIMIZATION

When using GA to solve problems, it is necessary to ensure a balance between the local and global search capabilities, which are determined by the crossover and mutation probabilities. GA achieves a balanced search capability by coordinating and competing through crossover and mutation operations, thus obtaining optimal solutions that consider both global and local searches. The term "coordination" refers to a population that is stuck in a hyperplane in the search space during evolution and is unable to escape crossover alone; mutation can help overcome this problem. The term "competition" refers the mutation operation destroying the expected building blocks formed by crossover. Similarly, in the iterative process of GA, adopting an elitist retention strategy can prevent the loss or destruction of the best individuals selected, crossed, and mutated during the iteration, preserve excellent individuals from the parent population to the offspring population, accelerate the convergence speed of the algorithm, and retain good genes.

Thus, when solving the model, selecting different crossover and mutation probabilities and numbers of elites can result in varying degrees of disturbance and convergence speeds for the same problem. To solve the problem proposes in this study, different parameter selections are considered to examine the potential impact on the results, and suitable parameters for the model are chosen through comparison. The experimental parameter settings are as follow: population



(a) P_C**=0.9,** P_m**=0.01 Iterative diagram**

(b) P_C**=0.9,** P_m**=0.1 Iterative diagram**

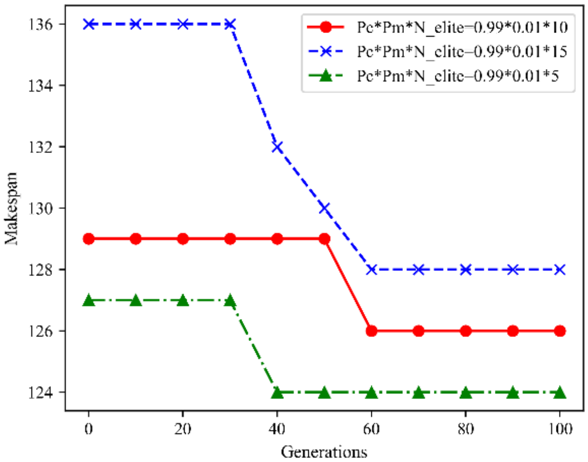
(c) P_C**=0.9,** P_m**=0.05 Iterative diagram**

(d) P_C**=0.95,** P_m**=0.01 Iterative diagram**

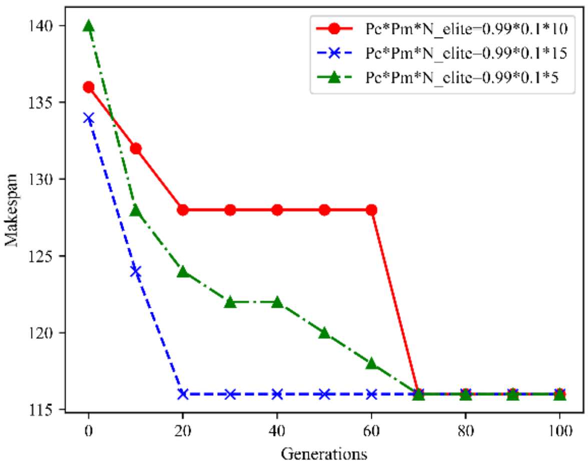
(e) P_C**=0.95,** P_m**=0.1 Iterative diagram**

(f) P_C**=0.95,** P_m**=0.05 Iterative diagram**

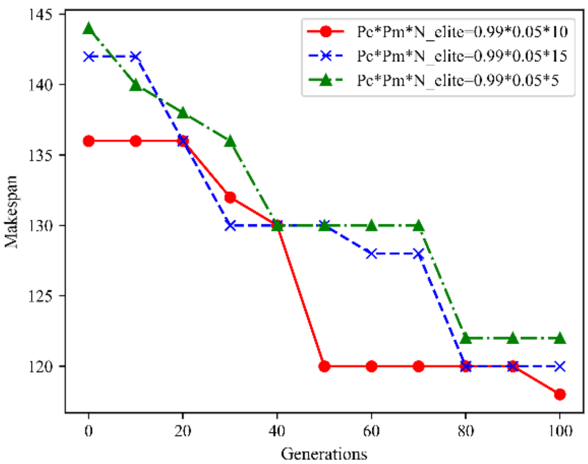
FIGURE 12. Comparison of iteration plots for different parameters.



(g) P_C=0.99, P_m=0.01 Iterative diagram



(h) P_C=0.99, P_m=0.1 Iterative diagram



(i) P_C=0.99, P_m=0.05 Iterative diagram

FIGURE 12. ***(Continued.)*** **Comparison of iteration plots for different parameters.**

size P = 100, maximum iteration number M = 100, crossover probabilities P_C of 0.9, 0.95, and 0.99, mutation probabilities P_m of 0.1, 0.05, and 0.01, and number of elites of 5, 10, and 15.

TABLE 3. Analysis and comparison of algorithm parameters.

Crossover probability	Mutation probability	Elite number	Optimal value	Iterat ions	Run time(s)
0.9	0.01	5	126	20	1.26
0.9	0.01	10	130	20	1.39
0.9	0.01	15	126	40	1.09
0.9	0.05	5	123	80	1.25
0.9	0.05	10	121	60	1.08
0.9	0.05	15	118	60	1.24
0.9	0.1	5	116	90	1.47
0.9	0.1	10	120	100	1.28
0.9	0.1	15	118	30	1.34
0.95	0.01	5	130	70	1.13
0.95	0.01	10	130	50	1.31
0.95	0.01	15	126	100	1.24
0.95	0.05	5	122	30	1.09
0.95	0.05	10	122	100	1.11
0.95	0.05	15	120	80	1.31
0.95	0.1	5	118	20	1.42
0.95	0.1	10	114	50	1.23
0.95	0.1	15	120	70	1.37
0.99	0.01	5	124	40	1.26
0.99	0.01	10	126	60	1.31
0.99	0.01	15	128	60	1.29
0.99	0.05	5	122	80	1.28
0.99	0.05	10	118	100	1.26
0.99	0.05	15	120	80	1.15
0.99	0.1	5	116	0	1.32
0.99	0.1	10	116	70	1.27
0.99	0.1	15	116	20	1.25

Figure.12 presents the best results obtained after running the model 20 times independently. Three sets of parameters were compared, each with three different options, resulting in a total of 3*3 = 27 combinations. Each of the figures shown in the graph represents only one variable and is analyzed and compared from two perspectives, as shown in the following Table 3.

The optimal solution of 114 is obtained when $P_C = 0.95$, $P_m = 0.1$, and elite=10, assuming that the other parameters remain unchanged. Although the fastest convergence speed is achieved with P_C (=0.9, $P_m = 0.01$, elite = 5) and ($P_C =$ 0.99, Pm =0.1, elite = 15), the corresponding optimal values are 126 and 116, respectively. The difference in the solution time between iterations is less than 1s. The parameters resulting in the optimal solution are chosen as follows: crossover probability $P_C = 0.95$, mutation probability $P_m = 0.1$, and number of elite individuals = 10. This parameter set has a relatively fast convergence speed and optimal results compared to the other parameters; thus, it is used to optimize the experiment in the following sections.

B. CASE SOLVING

This section provides a solution for the aforementioned cases, and the Gantt chart obtained from the experiment is shown in Figure 13.

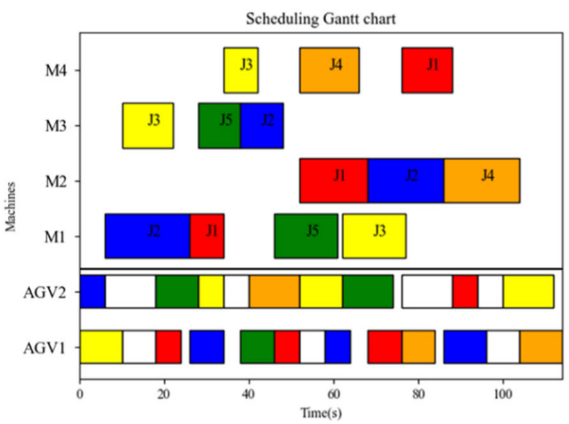


FIGURE 13. Example Gantt chart.

The Figure 13. shows the optimal scheduling scheme for the proposed example. The vertical axes represent the inspection machine and AGV, and the same rectangle represents different processes for the same workpiece. The colors displayes on the AGV represent the transportation of the corresponding workpiece, with white indicating that the AGV is currently idle. This follows the standard format uses in academic journals.

The inspection process of the sample rack in the scheduling plan shown in the figure is described as follows: The first inspection process of sample rack J_2 is arranged on inspection machine M_1. J_2 must be transported from the waiting area to inspection machine M_1. At this time, both AGV1 and AGV2 are idle. As AGV2 requires less time to transport the sample rack, it is selected to transport J_2. The second process of the sample rack J_2 can be inspected using machine M_3. Then the AGV transports it to M_3. At this time, machine M_3 inspects the sample rack J_5. The inspection completion time of J_2 at M_1 plus the transportation time by AGV is less than the inspection completion time of J_5. In addition, there is an idle AGV exists at this time; thus, AGV1 is scheduled to transport J_2 from M1 to M_3. After J_5 is processed, J_2 can begin the inspection. The transportation route of sample rack J_2 is $M_1 \to M_3 \to M_2$. The transportation route for AGV1 is from the waiting area to M_1; and the transportation route for AGV2 is M_1–M_3–M_2. The sample rack is placed in a refrigerator. The total inspection completion time of the sample rack is the sum of the first inspection time of 20 at M_1, the second inspection time of 18 at M_3, the third inspection time of 10 at M_2, the transportation time from the beginning to M_1 of 6, the transportation time of $M_1 \to M_3$ of 8, transportation time of $M_3 \to M_2$ of 6, and the transportation time from M_2 to the end of 10, which is $20 + 18 + 10 + 6 + 8 + 6 + 10 = 78$. The inspection process for subsequent sample racks followed the same procedure.

C. ALGORITHM COMPARISON

To verify and evaluate the performance of the GA-STW algorithm proposed in this study in solving the integration scheduling problem of AGVs and machines in sample-testing laboratories, a comparison is made with other algorithms that solve similar integration scheduling problems. The main comparison algorithms include six methods: the lower bound algorithm (LB) proposed by Ulusoy and Bilge [46], taboo search algorithm (MTS) proposed by Montane and Galvao [47], NSGA-II algorithm proposed by Ma et al. [26], improved iterative local search algorithm (ILS) proposed by Hu et al. [48], discrete whale optimization algorithm (IWOA) proposed by Zou et al. [49], and the original GA. The comparison data in this study are obtained from examples of AGV and machine integration scheduling in a flexible job shop proposed in [40]. The GA-STW algorithm proposes in this study is independently executed 25 times for each example to obtain the optimal solution. The results are compared with those of other methods reported in the literature, as shown in Table 4.

The Table.4 shows the optimal results obtained by the different algorithms in solving the test problems. The LB method proposed by Ulusoy yields a theoretically optimal solution. However, it is difficult to achieve this value in practical application. Comparing the 40 test cases in the table, it can be observed that the original GA has difficulty converging to the global optimal solution and exhibits poor performance. Compared with the results of the original GA, GA-STW proposes in this study has better results in 19 test cases, with improvements of over 5% in eight test cases, including three test cases with improvements of 10%. Of the 40 test cases presented in the table, GA-STW algorithm results are not inferior to those of the other algorithms, with 20% of the examples performing better than the other algorithms, and 62.5% obtain good results as well as other algorithms. Thus, it can be concluded that the GA-STW algorithm has its own advantages in solving AGV and machine integration scheduling problems, similar to the other algorithms.

TABLE 4. Comparison of algorithm results.

Example	Optimal solution						
	LB	MTS	NSGA-II	ILS	IWOA	GA	GA-STW
EX11	72	96	96	96	96	96	96
EX21	86	100	105	104	100	105	100
EX31	81	99	105	105	99	99	99
EX41	62	112	117	86	112	117	112
EX51	60	87	89	87	87	87	**79***
EX61	96	118	121	128	118	128	118
EX71	76	111	120	119	111	119	111
EX81	146	161	170	169	161	161	161
EX91	93	116	121	120	116	116	**113***
EX101	124	150	153	150	146	150	150
EX12	66	82	83	83	82	83	**80***
EX22	76	76	81	76	76	76	76
EX32	75	85	89	85	85	89	85
EX42	60	87	69	69	87	87	69
EX52	54	69	98	100	69	69	69
EX62	86	96	88	93	95	98	**86***
EX72	74	79	142	150	79	79	79
EX82	140	151	154	153	146	157	**142***
EX92	91	102	137	139	102	102	102
EX102	114	135	116	116	135	135	116
EX13	64	84	86	86	84	84	**79***
EX23	82	86	86	86	86	86	86
EX33	77	86	91	91	86	86	86
EX43	58	89	76	95	89	89	89
EX53	52	76	104	75	74	76	76
EX63	88	103	94	104	103	103	103
EX73	76	84	143	88	83	84	84
EX83	142	153	153	143	153	153	153
EX93	93	105	105	105	105	105	**95***
EX103	116	137	143	143	137	137	137

TABLE 4. *(Continued.)* Comparison of algorithm results.

Example	LB	MTS	NSGA-II	ILS	IWOA	GA	GA-STW
EX14	68	103	105	105	103	105	103
EX24	84	108	114	116	108	108	108
EX34	81	111	114	116	111	111	**107***
EX44	62	126	126	126	121	126	126
EX54	56	96	97	99	96	99	96
EX64	90	120	121	123	119	120	120
EX74	76	127	128	136	126	127	127
EX84	148	163	167	163	163	167	163
EX94	91	122	124	125	120	122	120
EX104	120	158	164	171	157	158	158

D. ALGORITHM UNIFORMITY COMPARISON

To validate the uniformity of the GA-STW algorithm proposed in this study, the aforementioned five algorithms, excluding LB and the GA-STW algorithm, are repeatedly executed for the case study EX51 fifteen times. A comparative analysis is conducted on the discrete distribution of the optimal solutions computed by the algorithms.

According to the calculated results in Table 5, the average value of the optimal solutions obtained from the six algorithms is computed. Additionally, the frequency of obtaining the optimal solution and the standard deviation are also determined. The results are presented in Table 6.

According to Table 6, the proposed GA-STW algorithm has a smaller average value of 84.5, whereas NSGA-II has the largest average value. GA-STW outperforms the original GA in terms of obtaining the most optimal solutions, demonstrating the effectiveness of the algorithm, with a probability of achieving a global optimum of 46.7%. The standard deviation comparison shows that GA-STW $<$ NSGA-II $<$ IWOA $<$ MTS $<$ GA $<$ ILS. Based on these three data analyses, it can be observed that the proposed GA-STW algorithm avoids the issue of the original GA becoming trapped in local optima, exhibiting good stability and optimization capabilities. Figure 14 is plotted based on the calculation data from Table 5, providing a visual comparison of the stability of the six algorithms during the computation.

Figure 15 presents a box plot generated based on the results obtained from the algorithmic solution of the case study in Table 5. The box plot offers the advantage of accurately and reliably describing the discrete distribution of data, while mitigating the influence of outliers. Each box represents the overall performance of the six algorithms and displays the maximum value, minimum value, median, and quartiles of the dataset.

TABLE 5. Comparison of algorithm solution results for EX51.

Number of experiments	Optimal solution					
	MTS	NSGA-II	ILS	IWOA	GA	GA-STW
1	89	89	87	92	92	**85***
2	83	**81***	87	**81***	87	**81***
3	91	**83***	91	89	88	88
4	**81***	88	80	85	85	**81***
5	84	**80***	**80***	**80***	83	82
6	87	89	87	87	87	**79***
7	89	86	**85***	89	94	86
8	85	85	91	**79***	90	83
9	83	88	**80***	**80***	84	84
10	**84***	92	91	86	90	86
11	**83***	87	**83***	**83***	85	**83***
12	81	89	80	92	83	**79***
13	82	**79***	89	**79***	92	84
14	88	**81***	89	91	83	83
15	84	93	83	84	86	**79***

TABLE 6. Stability analysis of six algorithms.

Algorithm	MTS	NSGA-II	ILS	IWOA	GA	GA-STW
Average value	85.5	88.6	86.8	88.3	87.3	84.5
Number of optimal solutions	3	5	4	6	0	7
Standard deviation	3.2	2. 5	4.1	3	3.6	2

According to Figure 15, it can be observed that the GA-STW algorithm in this study has the lowest median, indicating the smallest median value. Additionally, the area of the box plot is relatively small, which demonstrates that the algorithm produces better results when solving this type of problem. Moreover, the area of the box plot represents the size of the range, and GA-STW algorithm has the smallest area, indicating the smallest range. This confirms that the algorithm possesses strong stability.

E. CONVERGENCE VERIFICATION OF ALGORITHMS

The iterative curves of the six algorithms are shown in Figure 16. It can be observed that NSGA-II, ILS, and MTS

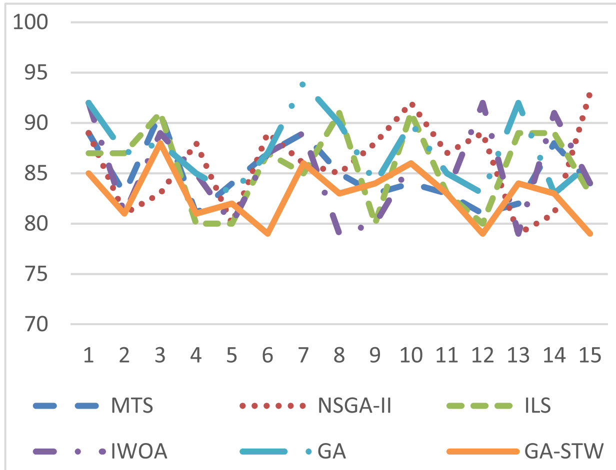


FIGURE 14. Example line chart of experiment results.

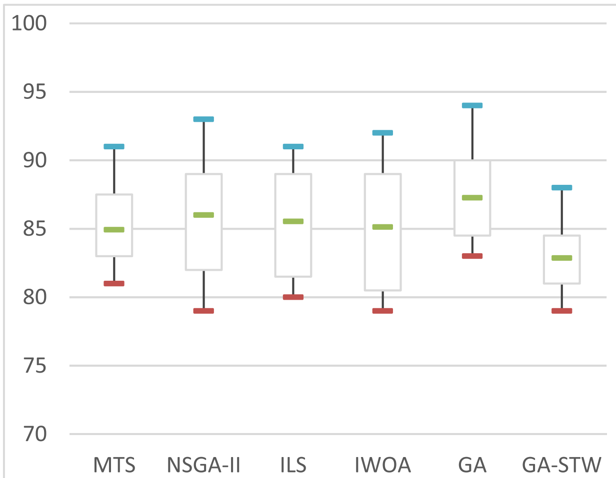


FIGURE 15. Six algorithm box charts.

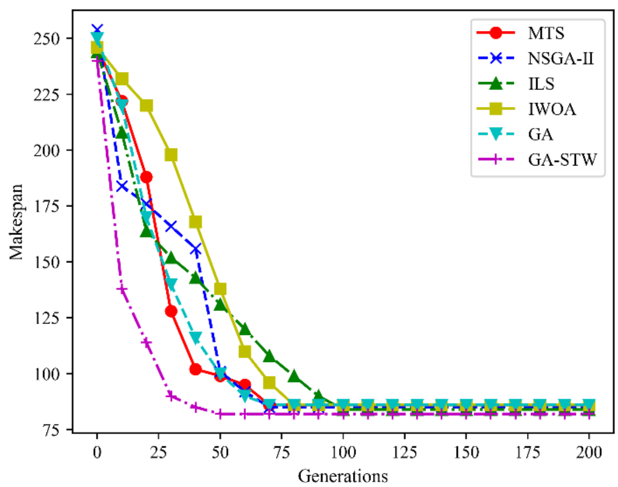


FIGURE 16. Algorithm convergence curve comparison chart.

exhibit faster speed and stronger capability in the early stages of the search. However, their speed slows down after a certain number of iterations, especially ILS, whose convergence speed gradually decreases in the later iterations. Both IWOA and GA maintain almost constant search speeds, with IWOA demonstrating stronger local search capability and a higher possibility of finding the optimal solution. GA shows better in the later stages of the search. The GA-STW proposed in this paper converges starting from the 50th generation and exhibits a fast convergence speed, yielding superior results. This demonstrates that in solving this type of problem, GA-STW presented in this study possesses excellent computational capability.

V. CONCLUSION

This study proposes a GA-STW algorithm to address the integration and scheduling issues between AGVs and inspection machines in sample-testing laboratories. The algorithm establishes a corresponding mathematical model to minimize the maximum TAT. GA is used to perform double-layer encoding of the process and the machine sequences during the solution process. The VAA is used to find the AGV with the shortest transportation time during decoding. To avoid idle time windows, the STW heuristic algorithm is used to consider the earliest inspection time of the sample and the idle status of the AGV, using a STW to minimize the maximum TAT. By optimizing the algorithmic parameter combinations, the most suitable parameter combination is selected to solve the test cases. A comparative analysis is conducted between the GA-STW algorithm and other algorithms using standard test cases. Additionally, the uniformity and convergence of the algorithm are verified through calculations on the corresponding test cases, thus demonstrating the applicability of the GA-STW algorithm to the mathematical model proposed in this paper.

With the continuous advancements in big data and technology, in future research, the scope of aspects that can be studied in a sample testing laboratory is broader as it presents a new background and new problems. For instance, considerations may include testing machine malfunctions, handling of emergency samples, and retesting of samples. Furthermore, the sample testing laboratory is not an independent department, and exploring its integration with other departments will be an intriguing research direction. Simultaneously, to address such problems, it is desirable to enhance the performance of the GA-STW algorithm, and introducing reinforcement learning or deep learning may yield superior results.

REFERENCES

[1] C. Yeo and W. Ng, "Automation and productivity in the clinical laboratory: Experience of a tertiary healthcare facility," *Singap. Med. J.*, vol. 59, no. 11, pp. 597–601, Nov. 2018.
[2] M. Tong and D. C. Ly, "Enhancing laboratory efficiency with total laboratory automation," *J. Clin. Chem. Lab. Med.*, vol. 3, no. 7, pp. 99–102, Feb. 2019.
[3] C. W. Lam and E. Jacob, "Implementing a laboratory automation system: Experience of a large clinical laboratory," *J. Lab. Autom.*, vol. 17, no. 1, pp. 16–23, Feb. 2012.
[4] R. S. Seaberg, R. O. Stallone, and B. E. Statland, "The role of total laboratory automation in a consolidated laboratory network," *Clin. Chem.*, vol. 46, no. 5, pp. 751–756, May 2000.
[5] L. Sarkozi, E. Simson, and L. Ramanathan, "The effects of total laboratory automation on the management of a clinical chemistry laboratory. Retrospective analysis of 36 years," *Clinica Chim. Acta*, vol. 329, nos. 1–2, pp. 89–94, Mar. 2003.
[6] S. E. F. Melanson, N. I. Lindeman, and P. Jarolim, "Selecting automation for the clinical chemistry laboratory," *Arch. Pathol. Lab. Med.*, vol. 131, no. 7, pp. 1063–1069, Jul. 2007.
[7] J. L. Allinson, K. E. Blick, and L. Cohen, "Ask the experts: Automation: Part I," *Bioanalysis*, vol. 5, no. 16, pp. 1953–1962, Aug. 2013.
[8] T. Yang, T.-K. Wang, V. C. Li, and C.-L. Su, "The optimization of total laboratory automation by simulation of a pull-strategy," *J. Med. Syst.*, vol. 39, no. 1, pp. 1–12, Jan. 2015.
[9] C. Archetti, A. Montanelli, D. Finazzi, L. Caimi, and E. Garrafa, "Clinical laboratory automation: A case study," *J. Public Health Res.*, vol. 6, no. 1, pp. 31–36, Apr. 2017.
[10] A. Caprara, P. Toth, D. Vigo, and M. Fischetti, "Modeling and solving the crew rostering problem," *Oper. Res.*, vol. 46, no. 6, pp. 820–830, Dec. 1998.
[11] O. M. Peck-Palmer, "Total lab automation takes teamwork," *Med. Lab. Observer*, vol. 41, no. 10, pp. 1–5, Oct. 2009.
[12] Y. P. Tabak, L. Vankeepuram, G. Ye, K. Jeffers, V. Gupta, and P. R. Murray, "Blood culture turnaround time in U.S. acute care hospitals and implications for laboratory process optimization," *J. Clin. Microbiol.*, vol. 56, no. 12, Dec. 2018, Art. no. e00500.
[13] A. L. H. Mayne, E. S. Mayne, and S. Louw, "Proposed application of six sigma metrics using a gamma distribution to monitor turnaround time for a high-volume coagulation test before and after introduction of a total laboratory automation platform in an academic laboratory in South Africa," *Int. J. Lab. Hematol.*, vol. 42, no. 3, pp. e124–e127, Jun. 2020.
[14] M. L. Yarbrough, W. Lainhart, A. R. Mcmullen, N. W. Anderson, and C.-A.-D. Burnham, "Impact of total laboratory automation on workflow and specimen processing time for culture of urine specimens," *Eur. J. Clin. Microbiol. Infectious Diseases*, vol. 37, no. 12, pp. 2405–2411, Dec. 2018.
[15] J. M. Baron, "Artificial intelligence in the clinical laboratory," *Clinics Lab. Med.*, vol. 43, no. 1, pp. 1–16, Mar. 2023.
[16] B. Wu, Y. C. Ding, and A. Basri, "Research status of integrated scheduling problem of automatic guide vehicle and machine," *Comput. Eng. Appl.*, vol. 59, no. 6, pp. 1–15, Jul. 2022.
[17] C. Z. He, Y. C. Song, and Q. Lei, "Integrated scheduling of multiple automated guided vehicles and machines in flexible job shop," *China Mech. Eng.*, vol. 30, no. 4, pp. 438–447, Feb. 2019.
[18] X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo, "Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems," *IEEE Access*, vol. 7, pp. 74909–74924, 2019.
[19] Y. J. Zhou, Y. C. Song, and Q. K. Wang, "Cluster genetic algorithm for multi-objective integrated scheduling of automatic guided vehicles and processing equipment," *China Mech. Eng.*, vol. 33, no. 1, pp. 97–108, Apr. 2021.
[20] X. Deng, X. B. Hu, and D. Y. Jiang, "Flexible job shop machine and AGV planning based on hybrid genetic algorithm," *J. Sichuan Univ.*, vol. 58, no. 2, pp. 73–82, Mar. 2021.
[21] X. X. Li, D. M. Yang, and X. Li, "Fusion scheduling method of flexible job shop automatic guide vehicle based on hybrid genetic whale optimization algorithm," *China Mech. Eng.*, vol. 32, no. 8, pp. 938–950, Sep. 2021.
[22] Z. F. Li and Y. Y. Liu, "Consider the job shop scheduling problem of charged multi-AGVs," *Comput. Integr. Manuf. Syst.*, vol. 27, no. 10, pp. 2872–2879, Oct. 2021.
[23] C. Liu, C. R. Zhang, and Y. X. Sun, "Application of improved adaptive genetic algorithm in multi-load AGV scheduling," *Mini Microcomput. Syst.*, vol. 42, no. 11, pp. 2241–2245, Nov. 2021.
[24] Y. H. Ge, S. Z. Li, and W. F. Li, "Research on collaborative scheduling of customized workshop production and logistics," *J. Nanjing Univ. Sci. Technol.*, vol. 45, no. 6, pp. 692–699, Dec. 2021.
[25] X. Wu, X. Wu, and P. H. Lou, "Anti-deadlock task scheduling of multi-load automatic guided vehicle system considering task travel time," *China Mech. Eng.*, vol. 32, no. 23, pp. 2840–2849, Dec. 2021.
[26] Q. H. Ma, X. L. Liang, and X. Y. Liu, "Research on multi-objective flexible job shop scheduling based on multi-AGVs and machine integration," *Comput. Eng. Appl.*, vol. 59, no. 1, pp. 278–290, Oct. 2023.
[27] S. M. Homayouni and D. B. M. M. Fontes, "Production and transport scheduling in flexible job shop manufacuring systems," *J. Global Optim.*, vol. 79, pp. 463–502, Feb. 2021.

[28] Y. J. Zou, Y. C. Song, and X. K. Wang, "Clustering genetic algorithm for multi-objective integrated scheduling of AGVs and machine," *China Mech. Eng.*, vol. 33, no. 1, pp. 97–108, Apr. 2022.
[29] M. H. Yuan, L. Z. Zhang, and K. J. Zhou, "Research on flexible shop scheduling based on improved hybrid genetic," *Ind. Eng. Manag.*, vol. 26, no. 6, pp. 95–103, Jan. 2021.
[30] M. Dai, D. Tang, A. Giret, and M. A. Salido, "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints," *Robot. Comput.-Integr. Manuf.*, vol. 59, pp. 143–157, Oct. 2019.
[31] M. Li and D. 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, Aug. 2021, Art. no. 104307.
[32] H. X. Qin, Y. Y. Han, Y. P. Liu, J. Q. Li, Q. K. Pan, and X. Han, "A collaborative iterative greedy algorithm for the scheduling of distributed heterogeneous hybrid flow shop with blocking constraints," *Exp. Syst. Appl.*, vol. 201, Sep. 2022, Art. no. 117256.
[33] H. X. Qin, Y. Y. Han, B. Zhang, L. L. Meng, Y. P. Liu, Q. K. Pan, and D. W. Gong, "An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 100992.
[34] H. X. Qin, Y. Y. Han, Y. T. Wang, Y. P. Liu, J. Q. Li, and Q. K. Pan, "Intelligent optimization under blocking constraints: A novel iterated greedy algorithm for the hybrid flow shop group scheduling problem," *Knowl.-Based Syst.*, vol. 258, Dec. 2022, Art. no. 109962.
[35] X. X. Li, D. M. Yang, and X. Li, "Flexible job shop AGV fusion scheduling method based on HGWOA," *China Mech. Eng.*, vol. 32, no. 8, pp. 938–950, Sep. 2021.
[36] L. He, R. Chiong, W. Li, G. S. Budhi, and Y. Zhang, "A multiobjective evolutionary algorithm for achieving energy efficiency in production environments integrated with multiple automated guided vehicles," *Knowl.-Based Syst.*, vol. 243, May 2022, Art. no. 108315.
[37] E. H. Liu, X. F. Yao, and T. Tao, "Improved flower pollination algorithm for job shop scheduling problem integrated with AGVs," *Comput. Integr. Manuf. Syst.*, vol. 25, no. 9, pp. 2219–2236, Aug. 2019.
[38] Y. P. Liu, M. Li, and C. Wang, "Investigation and analysis of laboratory turnover time of routine blood samples in emergency hospital," *Lab. Med. Clin. Pract.*, vol. 12, no. 5, pp. 611–613, Mar. 2015.
[39] X. W. Huang, S. F. Chen, and D. Y. Zhou, "A review of genetic algorithms for flexible job-shop scheduling," *Comput. Integr. Manuf. Syst.*, vol. 28, no. 2, pp. 536–551, Jul. 2022.
[40] Ü. Bilge and G. Ulusoy, "A time window approach to simultaneous scheduling of machines and material handling system in an FMS," *Oper. Res.*, vol. 43, no. 6, pp. 1058–1070, Dec. 1995.
[41] T. Murata, H. Ishibuchi, and H. Tanaka, "Genetic algorithms for flowshop scheduling problems," *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 1061–1071, Sep. 1996.
[42] G. Rudolph, "Convergence of evolutionary algorithms in general search spaces," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 50–54.
[43] K. D. Jong, "On using genetic algorithms to search program spaces," *Genetic Algorithms Their Appl.*, pp. 210–216, Jan. 1987.
[44] Y. Chen, A. Jaekel, and A. Bari, "Resource allocation strategies for a non-continuous sliding window traffic model in WDM networks," in *Proc. 6th Int. ICST Conf. Broadband Commun., Netw., Syst.*, 2009, pp. 1–7.
[45] E. F. Guedes, A. M. da Silva Filho, and G. F. Zebende, "Detrended multiple cross-correlation coefficient with sliding windows approach," *Phys. A, Stat. Mech. Appl.*, vol. 574, Jul. 2021, Art. no. 125990.
[46] G. Ulusoy and Ü. Bilge, "Simultaneous scheduling of machines and automated guided vehicles," *Int. J. Prod. Res.*, vol. 31, no. 12, pp. 2857–2873, Dec. 1993.
[47] F. A. T. Montané and R. D. Galvão, "A Tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," *Comput. Oper. Res.*, vol. 33, no. 3, pp. 595–619, Mar. 2006.
[48] X. Y. Hu, X. F. Yao, and P. Huang, "An improved iterative local search algorithm for flexible job shop scheduling problem with multiple AGVs," *Comput. Ind. Eng.*, vol. 28, no. 7, pp. 2198–2212, Jul. 2022.
[49] Y. J. Zhou, Y. C. Song, and Y. Wang, "Integrated AGV and machine scheduling method based on discrete whale optimization algorithm," *J. Chongqing Univ.*, vol. 45, no. 6, pp. 55–74, Apr. 2022.

HONGTAO TANG received the B.S. degree in chemical equipment and machinery and the Ph.D. degree in mechanical engineering from Zhejiang University, Hangzhou, Zhejiang, China, in 1998 and 2004, respectively.

He is currently an Associate Professor with the Institute of Industrial Engineering, Zhejiang University of Technology. His research interests include design and optimization of production and logistics system in factory and digital twin technology of production and logistics systems.

JIAWEI WANG received the B.S. degree in industrial engineering from the Shandong Institute of Business and Technology, Yantai, Shandong, China, in 2020. She is currently pursuing the M.S. degree in mechanical engineering with the Zhejiang University of Technology.

Her research interests include integration and scheduling of AGVs and machines.

ZHIXIONG YANG received the M.S. and Ph.D. degrees in mechanical engineering from Zhejiang University, Hangzhou, Zhejiang, China, in 2001 and 2005, respectively.

He is currently a Senior Technical Consultant with Zhejiang Jiaxing Sudoku Bridge Technology Company. His research interests include big data applications in enterprise digital transformation, data mining, and machine learning algorithms.