sustainability

MDPI

Article

Towards Sustainable Production: An Adaptive Intelligent Optimization Genetic Algorithm for Solid Wood Panel Manufacturing

Jingzhe Yang, Yili Zheng and Jian Wu *

School of Technology, Beijing Forestry University, Beijing 100083, China; yangjingzhe@bjfu.edu.cn (J.Y.); zhengyili@bjfu.edu.cn (Y.Z.)
* Correspondence: wujian@bjfu.edu.cn

Abstract: Optimizing production processes to conserve resources and reduce waste has become crucial in pursuing sustainable manufacturing practices. The solid wood panel industry, marked by substantial raw materials and energy consumption, stands at the forefront of addressing this challenge. This research delves into production scheduling and equipment utilization inefficiencies, offering innovative solutions for the solid wood panel processing line aimed at achieving environmental sustainability and operational efficiency. The study is articulated through two main segments: (1) an exhaustive analysis and the development of a simulation system for the solid wood panel processing line, delineating all production elements and operational logic, furnished with a user-friendly simulation interface, and (2) a comprehensive evaluation and enhancement of various scheduling algorithms specific to the Flexible Job-Shop Scheduling Problem (FJSP) encountered in solid wood panel workshops. A significant leap forward is made with the introduction of the Adaptive Intelligent Optimization Genetic Algorithm (AIOGA), an evolved version of the standard Genetic Algorithm (GA) engineered for optimal scheduling within the solid wood panel processing line. AIOGA incorporates advanced features such as encoding strategy, population initialization, objective function setting, selection strategy, crossover operation, and mutation operation, demonstrating the methodological depth of the study. We applied AIOGA in a designed FJSP, and AIOGA substantially reduced the maximum completion time to 90 min. It evidenced an improvement of 39.60% over the conventional GA, enhancing the equilibrium of the equipment workload across the system. This research presents a multifaceted strategy to address the scheduling complications inherent in solid wood panel production and highlights the extensive applicability of adaptive intelligent optimization in diverse industrial settings. This study establishes a new paradigm in manufacturing optimization, underlining the valuable integration of sustainability and efficiency in production methodologies.

Keywords: flexible job-shop scheduling problem; improved genetic algorithm; digital twin; intelligent manufacturing; sustainable production; resource optimization; energy efficiency

Citation: Yang, J.; Zheng, Y.; Wu, J. Towards Sustainable Production: An Adaptive Intelligent Optimization Genetic Algorithm for Solid Wood Panel Manufacturing. *Sustainability* **2024**, *16*, 3785. https://doi.org/10.3390/su16093785

Academic Editors: Vladimíra Biňasová and Branislav Micieta

Received: 21 February 2024
Revised: 22 April 2024
Accepted: 28 April 2024
Published: 30 April 2024

1. Introduction

1.1. Problem Statement

The significance of Earth's ecosystems for the economy and sustainable development has become a prominent topic of discussion in the public sphere. The scientific community has quantified the socioeconomic benefits of adequately functioning ecosystems through ecosystem services (ES) [1]. The industrial landscape has significantly transformed from manual activities to advanced, automated systems [2].

With the advent of Industry 5.0 [3], an evolution from Industry 4.0, there is a heightened emphasis on social and ecological values, prioritizing the safeguarding of the planet's ecological well-being alongside industrial advancement [4]. This paradigm shift underscores the importance of sustainability and responsible resource management in manufacturing processes. By integrating sustainable practices into production systems, such as

reducing waste generation, optimizing energy consumption, and promoting eco-friendly materials, Industry 5.0 aims to minimize environmental impact and cultivate a more sustainable industrial ecosystem. This approach ensures the long-term viability of manufacturing operations and contributes to the broader goal of building a greener and more resilient society. However, solid wood panel manufacturing faces specific sustainability challenges despite these advancements. The current production methods often lead to the excessive use of raw materials, energy inefficiencies, and significant waste generation, all of which bear heavily on environmental sustainability. The goal of optimizing production while reducing environmental footprint has become ever more critical in ensuring the wood panel industry's contribution to a sustainable future.

Simultaneously, the high demand for wood-based items has raised concerns about environmental matters [5]. Nevertheless, modern society needs help with the most effective utilization and productive manufacturing of wood-based items. Addressing the pressing need for sustainable production practices in the solid wood panel industry, this study investigates the applicability of evolutionary algorithms to enhancing resource efficiency, minimizing waste, and optimizing energy utilization—foundational pillars in the pursuit of ecological sustainability. In this context, evolutionary algorithms serve as a potent tool to bridge the gap between complex production scheduling challenges, and the sustainability goals within the solid wood panel industry. By leveraging advanced computation to unravel the intricacies of production logistics, these algorithms facilitate the creation of scheduling solutions that optimize efficiency while prioritizing environmental stewardship. The deployment of such algorithms in solid wood panel manufacturing drives operational excellence and supports sustainable practices by minimizing its ecological footprint and enhancing the use of renewable resources.

Wooden items have a profound connection to people's lives. With the development of people's preferences and spending power, the demand for wooden products is growing and buyers are becoming more discerning. These products require both unique design and affordable pricing. Consequently, the primary focus in the wood-based panel market has shifted from labor cost competition to competition based on product innovation, quality, pricing, delivery, and production expenses. Reducing the time it takes to supply goods will directly impact the profitability of a product [6,7].

The new framework of Industry 4.0, known as Reference Architecture Model Industry 4.0, aims to fully digitize the industrial shop floor, according to the German committees DIN and DEK [8]. To achieve the aforementioned goals of reducing product delivery time and increasing product profitability, contemporary production scheduling technology is employed to strategize and allocate resources inside the workshop production line [9,10].

The application of intelligent manufacturing in producing solid wood panels aims to achieve automation and efficiency in manufacturing. By embracing the principles of Industry 4.0, real-time data monitoring, intelligent production scheduling, and optimized resource utilization can be realized, leading to significant time and material savings [11]. This technology not only reduces waste and losses in the production process, but also enhances the flexibility and responsiveness of production lines, enabling companies to adapt more swiftly to changes in market demand, thereby bolstering competitiveness. By applying intelligent manufacturing technology, solid wood panel manufacturers can operate their production lines more efficiently, achieve cost reductions, and enhance product quality, resulting in substantial economic benefits [12].

1.2. Related Works

1.2.1. Characteristics of the Flexible Job-Shop Scheduling Problem (FJSP)

While there are a multitude of wood products, their production procedures are largely analogous. The processing sequence of a solid wood panel primarily consists of five fundamental processes: an outbound unit, machining, grinding, dedusting, and packaging unit. The procedure is not a linear activity, as at least one step will involve multiple pieces

of equipment. Thus, the challenge of scheduling the manufacturing of solid wood panels can be classified as a Flexible Job-Shop Scheduling Problem (FJSP) [13–15].

In a Flexible Job Shop, each workpiece can undergo a distinct sequence of operations, and each operation can be carried out using different machining resources [16]. The versatile workshop allows for easy adaptation to various production requirements, achieving high levels of efficiency and resource utilization. Key features of a flexible shop include non-uniform process pathways for workpieces, a wide range of machining equipment, and intricate task scheduling. The complexity of task scheduling in a flexible workshop arises from the wide range of workpieces and machining resources, necessitating the consideration of several parameters such as process routes, machining times, and resource limits.

FJSP is an extension of the traditional Job-Shop Scheduling Problem (JSP) [17]. In contrast to the conventional job-shop scheduling problem, flexible job-shop scheduling involves the use of processing equipment that is not predetermined but rather unknown for each operation. Specifically, the machine can be chosen for each step of every workpiece [18]. This leads to an escalation in the intricacy of scheduling and necessitates the development of efficient algorithms to address such issues.

1.2.2. Computational Intelligence in FJSP

In the realm of computational intelligence applied to FJSP, commonly utilized algorithms are broadly categorized into heuristic algorithms and emergently developed machine learning algorithms. Among heuristic approaches, prevalent algorithms include the Genetic Algorithm (GA) [19], Simulated Annealing (SA) [20], Tabu Search (TS) [21], Particle Swarm Optimization (PSO) [22], and Ant Colony Optimization (ACO). GA is recognized for its emulation of natural selection, which is frequently employed in addressing scheduling dilemmas. SA, introduced by Metropolis et al. [23], adopts a probabilistic decay strategy to target global optima, wherein its efficacy in combinatorial optimization for scheduling was demonstrated by Kirkpatrick et al. [24]. Initially proposed by Glover [25], TS leverages a taboo list to circumvent repetitive exploration, evidencing its capability in combinatorial optimization. PSO, formulated by Kennedy and Eberhart [22], is a population-based optimization strategy designed to identify optimal solutions through collective particle cooperation and learning. An improved PSO variant was introduced by Kennedy et al. [26] for scheduling in solid wood panel processing, adjusting the inertia weight and acceleration constants. Inspired by ant foraging behaviors, ACO deploys distributed computing for shortest-path discovery, leveraging pheromone communication mechanisms amongst ants, as elaborated by Marco Dorigo [27].

In the burgeoning field of machine learning algorithms, Artificial Neural Networks (ANN), depicted by Hopfield and Tank [28] as mimicking cerebral learning and processing, demonstrate optimization proficiency. Deep Learning (DL), with its capability for high-dimensional and complex data feature extraction, was extensively discussed by LeCun et al. [29]. In scheduling optimization, DL facilitates automated feature engineering within decision-making processes.

The studies referenced above illuminate that GA exhibits robust global search capabilities and a high degree of flexibility, with an algorithmic procedure that is relatively simpler than that of ANN and DL. However, during the search process, it is susceptible to premature convergence to local optima rather than global ones. The performance of SA heavily depends on the annealing process parameters. In high-complexity problems like FJSP, improper parameter settings could lead to lengthy computations to find approximate solutions. TS demands carefully designed taboo list lengths and mechanisms for generating candidate solutions, which can become complex to manage in high-dimensional search spaces. PSO offers commendable parallel processing capabilities and is inclined towards the paths of global and individual optima, yet may also prematurely fall into local optima in high-dimensional search landscapes. ACO is proficient in solving combinatorial optimization problems but performs poorly within the high-dimensional settings typical of FJSP. ANN possesses powerful non-linear modeling capabilities that are suitable for

pattern recognition and other issues. However, it requires extensive data for model training and has a complex network structure, adding to the algorithm's temporal cost. DL excels in decision-making tasks but necessitates significant time investment to learn complex tasks. Considering the advantages and limitations of various algorithms, this study adopts GA as the foundational approach for solving FJSP due to its superior global search capabilities and the relative simplicity of its algorithmic flow. While GA is prone to premature convergence to local optima, the careful design and adjustment of hyper-parameters could effectively counteract this issue by enhancing population diversity and expanding the search domain, thus balancing the global and local search in the exploratory process.

1.2.3. Optimizing GA for FJSP: A Comprehensive Review of Hyper-Parameter Adjustments

Goldberg and Lingle [30] demonstrated the applications of GA in scheduling, with the essence of GA being iterative evolutions through encoding strategies, population initialization, selection, crossover, mutation, and others to derive optimal solutions. Zhang et al. [31] enhanced the scheduling performance in solid wood panel processing by incorporating an improved design of fitness functions into GA. Olympia et al. [32] analyzed the impact of population size on the performance of GA in the modeling of cultivation processes. The findings revealed that population size decisively affects the search space and the quality of the solutions addressed by GA.

The selection strategy is a pivotal operation in GA, responsible for choosing superior individuals to become parent candidates for the next generation, thus influencing the algorithm's problem-solving capability and efficiency [33]. Common selection strategies in GA include roulette wheel selection, tournament selection, rank selection, elitist strategy, random selection [34], truncation selection [35], random pairing selection [36], and local selection [37]. Roulette wheel selection, which employs the principle of the randomness of a roulette wheel, assigns a selection probability proportional to each individual's fitness [38]. This strategy is simple to implement but may introduce bias due to the presence of extreme fitness values. Tournament selection emulates the competitive selection process in biological evolution. In tournament selection, a certain number of individuals are randomly selected from the current population to compete and the winners are selected as parents for the next generation based on their fitness values [39], thus increasing the chances of selection without relying on the fitness ranking. Rank selection assigns selection probability based on the fitness ranking of individuals, allowing even those with lower fitness to be selected, thereby reducing the probability of premature convergence and avoiding scaling issues with fitness values [40]. The elitist strategy ensures that the fittest individuals in the population are directly passed on to the next generation without going through the selection, crossover, and mutation processes [41]. This method ensures that excellent solutions in terms of fitness are preserved and helps to accelerate the algorithm's convergence. Fogel et al. [33] introduced an adaptive selection strategy to balance exploration and exploitation effectively and prevent premature convergence to local optima. Akarsh Kumar et al. [42] investigated a novel adaptive mechanism for GA that dynamically adjusts the mutation rate through a group elite selection strategy, thereby enhancing the algorithm's overall search efficiency and solution quality. Goldberg and Deb [41] conducted a comparative analysis of various selection schemes used in GA, demonstrating each scheme's impact on the algorithm's performance.

Crossover operation simulates the genetic exchange process in biological evolution, where parts of genes are chosen from two or more parent individuals and crossed over to generate new offspring [43]. The goal of this crossover is to amalgamate the strengths of different individuals, producing offspring with an enhanced performance. Crossover operations in GA include single-point crossover [44], multi-point crossover [45], uniform crossover [46], and Partially Mapped Crossover (PMX) [47]. Single-point crossover, in which a randomly selected position splits two parent chromosomes into two segments that are then swapped to generate offspring, can damage beneficial gene combinations, especially when advantageous genes are distributed on both sides of the crossover point.

In FJSP, this might lead to the algorithm being trapped in local optima, causing premature convergence. Multi-point crossover by selecting multiple random points to split the chromosome into segments and alternating between them might sever valuable gene combinations and increase complexity due to more cutting points. In FJSP, excessive cuts could disrupt the dependency relationships between processes, resulting in lower-quality offspring. Uniform crossover, where the exchange on each gene position is independently and randomly chosen, can excessively disrupt the structure of parental chromosomes, causing the offspring to lose the overall characteristics of the parents. Such random crossover without the consideration of logical process relationships could lead to the emergence of illegal or suboptimal solutions. PMX selects a crossover point between two parental individuals and then, starting from the crossover point, partially exchanges gene segments between the two individuals. This method retains some information from the parental individuals and generates offspring with new combinations through crossover operations.

In FJSP, PMX might limit the diversity and exploratory potential of solutions. Bandaru et al. [48] optimized GA for single-objective optimization problems by modifying the SBX (Simulated Binary Crossover) and mutation operations, building in an adaptive mechanism. Srinivas and Patnaik [43] proposed an adaptive crossover and mutation rate for GA, and Hinterding et al. [49] confirmed the rationality and effectiveness of dynamic adaptive crossover rate strategies. Eiben et al. [50] reviewed various adaptive mechanisms. Semenkin and Semenkina [51] explored self-configuring GA with improved uniform crossover operations. Watanabe et al. [52] proposed a modified genetic algorithm with an adapted crossover operator and adaptive search area specifically designed for job-shop scheduling problems to improve the efficiency of solving scheduling issues. Viana et al. [53] likewise improved GA, combining local search strategies and multiple crossover operators for the job-shop scheduling problem, enhancing the quality of algorithm solutions and search efficiency and increasing the precision of job-shop scheduling problem solutions.

Mutation operation simulates the gene mutation process in biological evolution by randomly changing gene values in an individual's chromosome with a certain probability [54]. The purpose of mutation is to introduce new gene combinations, increase the diversity of the search space, and prevent the algorithm from being trapped in local optima. Standard mutation methods include bit-flip mutation, uniform mutation, and Gaussian mutation, etc. Bit-flip mutation is the most common mutation operation, randomly selecting a gene position and flipping its value [55]. This method is simple and effectively introduces randomness. However, its disadvantage is that, for problems with longer coding lengths or real number coding, this method may change too many or too few genes, causing large fluctuations in the quality of the solution. In FJSP, this may destroy the logical relationship between processes. Uniform mutation randomly changes each gene in an individual with a uniform probability [56]. This mutation method can introduce new genes throughout the search space. However, it may result in the loss of excellent gene combinations, potentially producing illegal or inefficient scheduling arrangements in FJSP. Gaussian mutation is suitable for real number coding by adding a small zero-mean Gaussian random number to the current gene value [57]. This method is suitable for fine-grained optimization, but tends to confine the algorithm to a local area, which may limit global search effects in FJSP. Scholars have made a series of improvements to mutation operations to further enhance the performance of GA in FJSP.

Matousek and Nolle [58] introduced a Hill Climbing (HC) mutation mechanism to improve GA by adding a Hill Climbing strategy and incorporating some local search strategies to optimize the search process and improve the search efficiency. Rajakumar [59] proposed an adaptive mutation technique and discussed the impact of static and adaptive mutation techniques on the performance of GA in the study, demonstrating the potential of adaptive mutation to improve efficiency. Khair et al. [60] analyzed the impact of mutation operations on GA when solving the Max One problem. The study aimed to compare and analyze the impact of various mutation strategies on algorithm performance. Neubauer [61] conducted a theoretical analysis of non-uniform mutation operations in modified GA. The

study aimed to improve the algorithm's ability to find global optima in combinatorial optimization problems through this mutation operation.

1.3. Our Contributions

Initially, the application layer of the solid wood panel processing system is established. This application layer facilitates simulation functionalities, allowing for inputting AGV speeds on the interface and conducting production simulations at corresponding speeds. It also enables simulations under varying workflows and quantities of processing equipment. This simulation system allows for testing the intelligent scheduling algorithm performance even before the actual production line (APL) is constructed, greatly enhancing the production sustainability. In reality, setting up a new production line entails high costs, whereas building a simulation system can significantly reduce production expenses and ensure sustainable production practices.

Secondly, after comparing various algorithms, this study employs an improved genetic algorithm to solve FJSP due to its unique characteristics. It enhances the production efficiency of the flexible job shop for solid wood panels and shortens the delivery time to the manufacturer. Moreover, it reduces production costs, improves the flexibility of the production line, optimizes resource utilization, and enhances the stability and reliability of the production line. This study provides new ideas for solving the processing of solid wood panels and new methods for the theoretical modeling of production scheduling. Crucially, our proposed Adaptive Intelligent Optimization Genetic Algorithm (AIOGA) directly addresses the sustainability aspirations of the solid wood panel industry by prioritizing resource optimization and waste minimization within the production scheduling framework. This not only catalyzes operational efficiency, but also ensures that the manufacturing process adheres to the principles of sustainable development, aligning economic gains with environmental stewardship.

2. Materials and Methods

2.1. The Construction of the Simulation System for Solid Wood Panel Processing

2.1.1. Analysis and Breakdown of the Solid Wood Panel Processing System Flow

The production line of solid wood panels is characterized by its complex structure, intricate interconnections, and diverse types of equipment. The solid wood panel processing procedure encompasses inspecting wood edges, retrieval from storage, machine processing, sanding, dust removal, packaging, and subsequent return to storage. This paper delves explicitly into the five main stages of the outbound unit, machining, grinding, dedusting, and packaging unit, as depicted in Figure 1.

The solid wood panels are placed on the pallets for edge inspection. When the inspection is completed, the robot arm grabs the pallets and the solid wood panels and places them in the outbound storage unit. The staging robot of the outbound storage unit places the solid wood panels on the staging table of the outbound storage unit. Automated Guided Vehicles (AGVs) play a crucial role in transporting materials between units in the solid wood panel processing line. AGVs are industrial vehicles designed to load goods automatically or manually and navigate to predetermined locations along a predefined route. At these locations, they can autonomously or manually handle the loading and unloading of goods. After that, the AGV takes the pallets from the staging table to realize the outbound storage and places them on the machine tools for processing. When the processing is completed, the AGV places the pallets in the sanding room. After grinding, the robot arm removes the pallet and places it in the dust room for dust removal. Finally, the AGV places the machined workpieces on the packaging table for packing.

To initiate the physical layer production line for processing solid wood panels, this study identifies the following key production elements as essential: Automated Guided Vehicles (AGVs), industrial robots, and stacker robots.

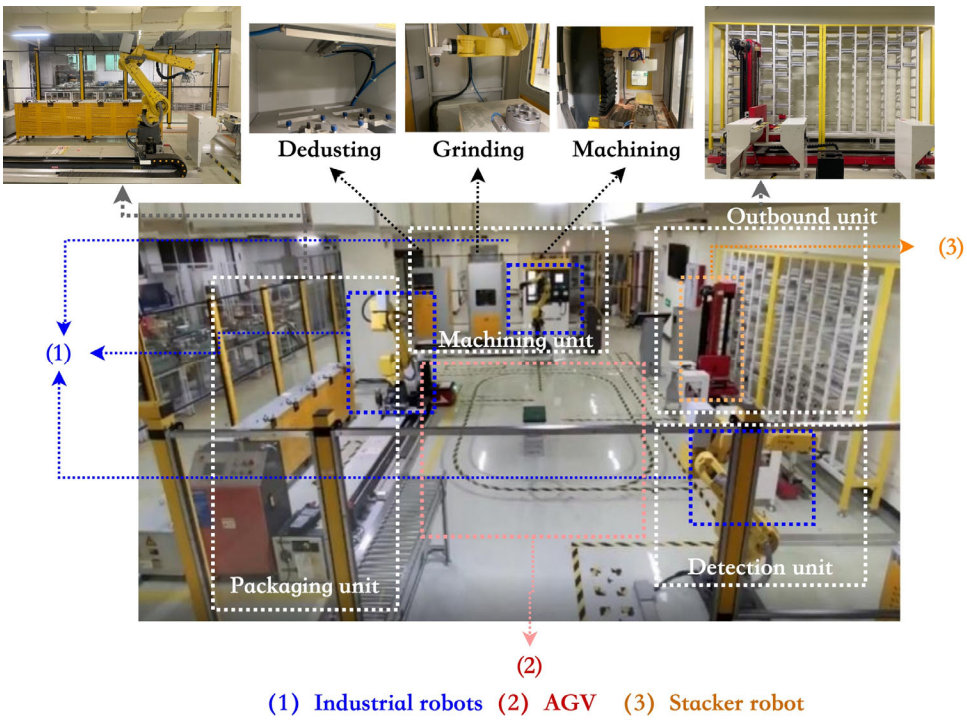

Figure 1. Main steps in the production of solid wood panels.

For this study, light-load-lifting AGVs from Zhongzhi Robotics Company were chosen, outfitted with magnetic navigation, and enhanced with safety features, including eight-point ultrasonic, infrared, and a safety touch edge. The actual image of the AGV is presented in Figure 2. The performance specifications are detailed in Table 1.

Figure 2. Image of AGV.

Table 1. Light-load-lifting AGV performance specifications.

Parameter Name	Light-Load Lifting AGV
Weight-carrying capacity	2–10 m
Positioning accuracy	5–10 m
Navigation accuracy	6–15 m
Driving method	Two-wheel differential drive
Navigation mode	Magnetic navigation

This study concentrates on two models of industrial robots: the M-20iA/35M and the M-10iD/12. Figure 3 provides images of these industrial robots. These robots feature advanced servo control technology, ensuring seamless integration and high-speed operation stability. The M-20iA/35M model's expansive reach radius of 1813 mm is particularly suited for outbound and packaging units, where an extended reach is essential for practical grasping tasks. On the other hand, the M-10iD/12 model is deployed within the machining unit, benefiting from its adequate reach radius of 1441 mm, suitable for the unit's operational requirements. The performance parameters are outlined in Table 2.

(**a**) (**b**)

Figure 3. (**a**) M-20iA/35M and (**b**) M-10iD/12.

Table 2. Industrial robot performance specifications.

Parameter Name	M-20iA/35M	M-10iD/12
Load capacity	35.0 kg	12.0 kg
Number of control axes	6-axis	6-axis
Reachable radius	1813 mm	1441 mm
Transportable mass	20 kg	12 kg
Repeat Positioning Accuracy	±0.03 mm	±0.02 mm
Industrial Robot Quality	250 kg	145 kg

In the outbound unit, stacker robots are used in conjunction with vertical storage systems to grasp and store materials on the racks. After evaluating different stacker robots, this study selected the Interlake Mecalux Custom Mini AS/RS Stacker Crane by Interlake Mecalux Company as the small stacker robot for operations with shelving systems. This compact stacker robot can handle vertical material while moving horizontally. An actual image of the stacker robot is depicted in Figure 4, and the performance specifications are itemized in Table 3.

Table 3. Stacker robot performance specifications.

Specification Parameter Name	Light-Load Lifting AGV
Lateral movement speed	60 m/min
Load capacity	15 kg
Positioning accuracy	±5 mm
Repeat positioning accuracy	±0.1 mm

Figure 4. Image of stacker robot.

2.1.2. Methodology for Constructing the Simulation System in Unity3D

Considering the impracticality of individually testing the physical layer for multiple production lines, this study introduces a viable solution by creating a simulation system within an application layer that can be tailored. Specifically for solid wood panel processing, this application layer is developed utilizing Unity3D. Upon importing the involved models into Unity3D, angles and coordinates are initialized to position the models centrally within the scene. The models' hierarchical relationships are simplified to facilitate subsequent mounting and coding. Afterward, the constructed geometric models are imported into the Unity3D environment for rendering treatment, enhancing the visual representation of the models to more closely approximate their real-world counterparts, thereby elevating the realism of the models to offer more precise and detailed visual effects.

Given that the entire processing logic is anchored around the operation of AGVs, this research takes the perspective of AGVs, reorganizing the logical sequence of components in Unity3D. Without considering the continuous production state in system downtime, the response of each unit triggered by the AGV within a cycle is segmented and displayed. Through this method, the operational logic of the solid wood panel processing line is clearly outlined and the simulation design of the entire system is realized at the application layer. The primary function of this layer is to interact with users, showcasing the operational results of the designed AIOGA within the simulation interface.

Within the Unity3D simulation environment, AGVs navigate predefined coordinates through path planning. Specifically, the built-in MoveTowards function of Unity3D is used to automatically guide the AGVs to each designated coordinate position, linking the motion speed parameters with the speed values set in the simulation interface. Upon the AGV reaching a predefined coordinate, a counter increments accordingly, serving as an action basis and signal for various units such as the inspection, storage retrieval, processing, and packaging units. This movement strategy applies equally to action subjects within the storage retrieval, processing, and packaging units—including industrial robots and stacker robots—whose motion mechanisms are also established following the AGV path planning guidance. Subsequently, the RotateTowards function is employed to execute the rotation of the industrial robots' axes, performing precise operations such as grasping and transferring. When AGVs carrying raw materials arrive at the processing unit, the industrial robots receive the movement signal, activate, and position themselves at the processing unit

staging area. Upon reaching the preset position, the industrial robots activate, execute material picking, and transfer them to the processing station. These industrial robots will perform subsequent actions based on new control signals as job requirements change.

2.2. Production Scheduling Framework for Solid Wood Panel Flexible Job Shop

The production process of the flexible workshop for solid wood panels entails the sequential processing of raw materials through various devices. The process comprises the processing of n workpieces through m stages to complete production. The duration of each process is known and can be observed in Figure 5.

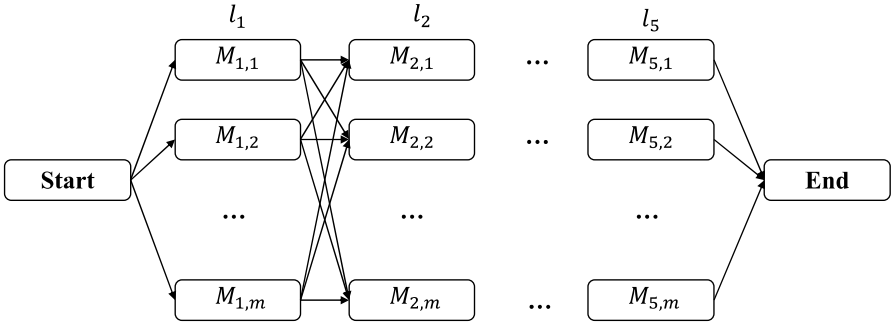

Figure 5. Problem description of production scheduling in a solid wood panel flexible job shop.

The set of workpieces is A:

$$A = J_m, m = \{1, 2, 3, \cdots, n\} \tag{1}$$

The collection of processes is denoted as B. The solid wood panel flexible job-shop production method consists of five primary production processes:

$$B = l_k, k = \{1, 2, 3, 4, 5\} \tag{2}$$

That means the elements l_1, l_2, l_3, l_4, and l_5 represent the five processes of the outbound unit, machining, grinding, dedusting, and packaging unit, respectively.

The set of machinable machines for a process i is denoted as M_i, where, for example, the set of all machinable machines for the first process is M_1. For the i process, the j machinable machine is denoted by $M_{i,j}$. For instance, if there are three selectable machines on the first process, the set of selectable machines on the first process is $\{M_{1,1}, M_{1,2}, M_{1,3}\}$.

It is assumed that the specified limitations are satisfied throughout the operation of this flexible job shop for solid wood panels:

- A solid wood panel can exclusively undergo processing on a specific equipment, following a specific procedure, at a particular moment.
- The equipment can process only one solid wood panel at a time.
- There exists a logical correlation and sequential limitations between the processing procedures of a solid wood panel.
- The processing of solid wood panels must be carried out continuously from the beginning to the end of the production process without any interruptions.
- The processing machinery remains operational without experiencing any malfunctions throughout the production process.
- The processing time utilized by the various processing machinery is identical for every process.
- Processes other than the five core processing steps need minimal time.

2.3. Adaptive Intelligent Optimization Genetic Algorithm (AIOGA)

As previously mentioned, GA is an optimization technique that mimics the processes of natural selection and genetic mechanisms. This algorithm employs an evolutionary pro-

cess to seek the most fitting solution for a given problem, utilizing mutation, crossover, and gene selection processes. GA is beneficial for complex optimization challenges, like FJSP, which is regarded as NP-hard due to the complications associated with task arrangement in a flexible manufacturing system with intricate combinatorial properties. Determining the specific workpiece for each task and the best-suited machine for each machining procedure is crucial. Adaptive Intelligent Optimization Genetic Algorithm (AIOGA) algorithm building on the genetic algorithm framework has been refined and tuned to enhance solutions' adaptability, flexibility, and quality. Initially, GA has the advantage in multi-point search, global exploration, and avoiding premature convergence; in the context of FJSP, GA can consider multiple solutions concurrently instead of a singular solution, effectively maintaining diversity during the search space exploration, thereby avoiding local optima. However, despite its advantages in multi-point search and global exploration, GA still needs help in parameter setting, intricate adjustment, and the propensity to fall into local optima. In highly complex issues like FJSP, GA struggles to balance maintaining diversity and search depth simultaneously [38]. To effectively address FJSP, this study presents the AIOGA, which introduces an adaptive mechanism for the dynamic tuning of parameters, adjusting the probabilities of crossover and mutation by monitoring the distribution of fitness and switching selection strategies based on different population states. This mechanism aids the algorithm in escaping local optima, efficiently manages various production demands, and enhances the production efficiency and flexibility while reducing production costs [18]. Figure 6 outlines the flowchart of AIOGA.

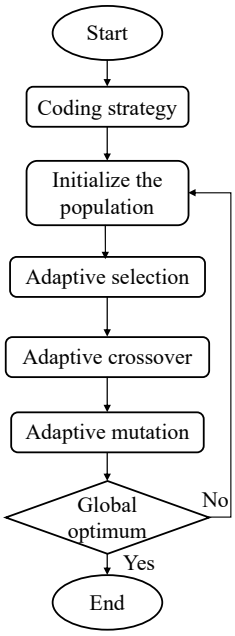

Figure 6. Flowchart of the improved genetic algorithm.

This research delineates AIOGA in five key aspects: coding strategy and population initialization [31], objective function, selection strategy [33], crossover operations [43], and mutation operations [54].

2.3.1. Coding Strategy and Initialize the Population

In this study, a hybrid encoding scheme and an effective population initialization method were designed for FJSP inspired by Zhang [31]. This hybrid encoding scheme integrates machine selection and operation sequencing into an efficient coding structure that directly maps solutions. The specific encoding strategies are as follows:

The machine sequence (M sequence) is represented as:

$$M = (m_1, m_2, \ldots, m_N), m_i \in Z^+, i = 1, 2, \ldots, N \quad (3)$$

Each integer m_i represents the machine number assigned to the ith operation. This sequence is directly related to specific processing tasks and forms the core of the scheduling plan.

The operation sequence (O sequence) is defined as:

$$O = (o_1, o_2, \ldots, o_N) \tag{4}$$

It consists of a series of real numbers o_1, o_2,..., o_N within the [0,1] interval, where each real number o_i denotes the execution priority of the ith operation within the entire manufacturing process. The relative magnitude of these numbers determines the order of operations.

$$0 \leq o_i \leq 1, i = 1, 2, \ldots, N \tag{5}$$

The chromosome can be represented as a tuple:

$$C = (M, O) \tag{6}$$

where M and O stand for the machine and operation sequences, respectively, and expanding the tuple yields:

$$C = (m_1, m_2, \ldots, m_N, o_1, o_2, \ldots, o_N) \tag{7}$$

This hybrid coding strategy effectively separates the two optimization dimensions of machine selection and execution order, facilitating GA to maintain the validity of the process routes and resource allocation during crossover and mutation [62]. Moreover, it enables the subsequent genetic operations to decode actual scheduling solutions directly.

Each chromosome in the generated initial population represents a potential solution and can be denoted by a two-dimensional array Φ, which contains the encoding information of all individuals:

$$\Phi = \begin{bmatrix} {m_1}^{(1)} & {m_2}^{(1)} & \ldots & {m_N}^{(1)} & {o_1}^{(1)} & {o_2}^{(1)} & \ldots & {o_N}^{(1)} \\ {m_1}^{(2)} & {m_2}^{(2)} & \ldots & {m_N}^{(2)} & {o_1}^{(2)} & {o_2}^{(2)} & \ldots & {o_N}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ {m_1}^{(p)} & {m_2}^{(p)} & \ldots & {m_N}^{(p)} & {o_1}^{(p)} & {o_2}^{(p)} & \ldots & {o_N}^{(p)} \end{bmatrix} \tag{8}$$

P denotes the population size, i.e., the number of chromosomes in the initial population. N represents the total number of operations in each chromosome.

${m_i}^{(p)}$ indicates the machine number assigned to the ith operation of the pth chromosome.

${o_j}^{(p)}$ signifies the position of the jth operation in the execution order of the pth chromosome, which is a value uniformly randomly generated within the [0,1] range.

This two-dimensional array structure represents the algorithm's first-generation population, allowing each individual's evaluation and genetic operations during the subsequent evolutionary process, efficiently exploring the search space for optimal solutions. This approach allows for the generation of an initial population with broad diversity, laying the groundwork for the smooth execution of the algorithm and optimizing its search efficiency and solution quality for specific problems while maintaining the universality of the algorithm.

2.3.2. Objective Function

In addressing FJSP, selecting an appropriate objective function is crucial for delineating the optimization direction of the solution algorithm, as it defines the endeavor's focal point [63]. In this study, we chose the makespan as the primary performance metric [63] to minimize the maximum completion time across all operations. The objective function can be formalized as:

$$\min T_M = \min\left(\max\left(T_{i,j}\right)\right) \tag{9}$$

Here, $T_{i,j}$ represents the completion time of the ith job on the jth machine, that is, the finishing time required by the equipment $M_{i,j}$, with T_M being the makespan for all jobs. The objective function's target is to reduce the duration of the overall operation completion time.

The makespan is not only linked to the production response speed, but it also indirectly reflects the throughput rate of the production flow. Lowering the makespan signifies the rapid completion of production tasks, impacting the business's responsiveness and adaptability in the marketplace. In scheduling optimization, the choice of makespan as the objective is particularly relevant, since it directly reflects the efficiency of the production process and aids in diminishing the overall operational costs [64]. Therefore, this research focuses on minimizing the makespan, concentrating efforts on the optimization of a singular metric, thus laying a solid foundation for subsequent discussions and conclusions, ensuring the practicality and effectiveness of the proposed algorithm.

2.3.3. Selection Operation

Inspired by Fogel et al. [33], this research introduces adaptive selection strategies to encourage an effective balance between algorithm exploration and exploitation while preventing premature convergence on local optima. These strategies leverage real-time population diversity and fitness metrics monitoring to modulate the selection process. Such adaptive selection, contingent on the population's live state, employs each selection mechanism's strengths to bolster the GA performance.

The adaptive strategy's mathematical model and decision-making criteria are delineated below:

(1) The population's genetic diversity index H is defined akin to the Shannon diversity index in ecology [65]:

$$H = -\sum_{i=1}^{s} p_i \log(p_i) \tag{10}$$

Here, p_i is the proportion of the ith allele within the population and s represents the total number of alleles.

(2) The adaptability of the population is gauged by calculating the average fitness $\overline{f}$ and the best fitness *fbest*:

$$\overline{f} = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{11}$$

$$fbest = \max\{f(x_i) | 1 \le i \le N\} \tag{12}$$

N is the population size and $f(x_i)$ denotes the fitness of chromosome x_i [38].

(3) Dynamic decision thresholds ΔH, $\Delta\overline{f}$, and $\Delta fbest$ are established. These thresholds are deployed to modulate the choice of selection methods, for instance:

$$\Delta H < \theta_H \tag{13}$$

$$\Delta\overline{f} < \theta_{\overline{f}} \tag{14}$$

$$\Delta fbest < \theta_{fbest} \tag{15}$$

where θ_H, $\theta_{\overline{f}}$, and θ_{fbest} act as preset thresholds for the variations in diversity, average fitness, and best fitness, respectively. For this study, such thresholds are configured as [46]:

$$\theta_H = 0.01 \tag{16}$$

$$\theta_{\overline{f}} = 0.05 \cdot \max\{f(x_i | 1 \le i \le N)\} \tag{17}$$

$$\theta_{fbest} = 0.01 \cdot \overline{f} \tag{18}$$

(4) A selection strategy function $S(t)$ is articulated based on the thresholds above, dictating strategy shifts under diverse evolutionary states:

$$S(t) = \begin{cases} roulette\ wheel\ selection, & if\ \Delta H > \theta_H \\ tournament\ selection, & if\ \Delta \overline{f} < \theta_{\overline{f}} \\ rank\ selection, & if\ \Delta fbest < \theta_{fbest} \\ elitist\ strategy, & others \end{cases} \tag{19}$$

When designing adaptive genetic algorithms, a salient technical issue is the trade-off between the diversity of the population's genetics and the algorithm's convergence rate. Dynamic adjustment of the selection strategy is imperative to maintain algorithmic performance in a complex search space.

Therefore, roulette wheel selection is adopted when the change in the population's diversity index H outstrips the preset threshold θ_H. This method considers the fitness of all individuals, furnishing probabilistic opportunities for selection, inclusive of individuals with lesser fitness. It is notably efficacious in upholding diversity and beneficial when the population is prone to premature convergence to avert the loss of superior solutions [66].

When the change in the average fitness $\overline{f}$ fails to meet $\theta_{\overline{f}}$, signifying a slow evolution pace in the population, tournament selection is chosen. This speeds up the inheritance of superior individuals by selecting the fittest within a randomly chosen subset, proving more efficacious during phases when heightened exploratory efforts are necessary [67].

Rank selection is initiated if the best fitness *fbest* alteration does not satisfy the preset threshold θ_{fbest}. Even if the population's overall average fitness slowly elevates, some individuals may be nearing optimal solutions. Rank selection ensures that these potentially superior individuals are accurately identified and continually optimized [41].

In other cases, an elitist strategy is employed, directly transmitting a subset of the fittest individuals in the current population to the next generation to maintain the genetic momentum of the optimal solutions discovered thus far [68].

In summation, the adaptive genetic algorithm framework dynamically fine-tunes itself based on the real-time state of the population, taking advantage of the benefits each selection method offers, grounded in stringent theoretical and empirical evidence. This method maintains genetic diversity amongst the population and prevents premature convergence during natural selection, thereby amplifying the algorithm's search efficiency and solution quality.

2.3.4. Crossover Operation

In this study, we introduced an adaptive crossover strategy to dynamically adjust the algorithm in response to the real-time evolutionary state of the population, integrating the advantages of various crossover techniques to enhance the algorithm performance. Our strategy draws inspiration from the research by Srinivas and Patnaik [43] on adaptive crossover and mutation rates in GA, influenced further by the work of Hinterding et al. [49] on the validity and efficacy of dynamic adaptive crossover strategies and a review of adaptive mechanisms by Eiben et al. [50].

This adaptive crossover operation considers the characteristics of FJSP, such as dependencies between operations and machine operation constraints. It dynamically adjusts the crossover strategy based on the population's current state and evolutionary trends, better preserving beneficial gene combinations and achieving a balance between global optimization and maintaining population diversity. The specific implementation steps are as follows:

(1) An initial crossover rate is established as the baseline value for crossover operations.

$$p_{c0} = 0.8 \tag{20}$$

(2) To assess fitness improvement, the current population's average fitness $\overline{f}$ is calculated at the end of each generation, and the improvement in the average fitness Δfit is obtained by a comparison with the previous generation:

$$\Delta fit = \overline{f}_{current} - \overline{f}_{previous} \tag{21}$$

(3) Two adaptive factors, α and β, where $\alpha > 1$ serves to increase the crossover rate and $\beta < 1$ serves to decrease it, are defined. The crossover rate p_c is dynamically adjusted based on the magnitude of Δfit, with the absolute threshold set to 0.01, following the research outcomes of K. DeJong and others [69,70]. A population improvement exceeding 1% signifies a significant advancement and below 1%, a modest progression.

If $\Delta fit < 0.01$, indicating slow population progress, the crossover rate is increased to generate more novel individuals:

$$p_c = \min(1, p_c \cdot \alpha) \tag{22}$$

Conversely, if $\Delta fit \geq 0.01$, suggesting rapid population optimization, the crossover rate is decreased to maintain the quality of individuals:

$$p_c = \max(p_{clb}, p_c \cdot \beta) \tag{23}$$

where p_{clb} is the lower bound of the crossover rate, preventing the search process from stagnation.

(4) Ensuring the crossover rate p_c stays within pre-set upper and lower limits, Eiben et al. [71] highlighted that too low of a crossover rate could diminish the algorithm's capacity to explore new solutions. In contrast, a high rate might disrupt existing quality solutions, affecting the algorithm's convergence speed. Moreover, classic studies by De Jong [69] suggest that a crossover rate between 0.6 and 0.9 yields the best performance for GA, especially in handling complex optimization tasks. Recent research by Srinivas and Patnaik [43] further confirms that this range enhances the algorithm's flexibility and adaptability. Thus, in this research, the fluctuation range for the crossover rate is set to [0.6, 0.9].

During the reproduction of the next generation, crossover operations are conducted with the adjusted crossover rate p_c. The flowchart for the adaptive crossover operation is illustrated in Figure 7.

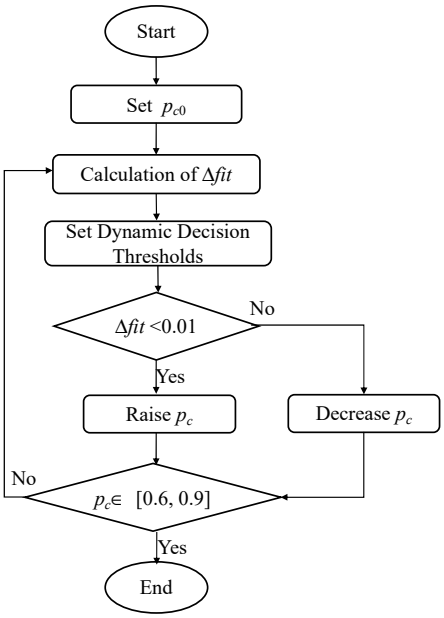

Figure 7. The flowchart of the adaptive crossover operation.

This adaptive crossover strategy aims to improve the algorithm's adaptability at different evolutionary phases, optimizing its capability to search for high-quality solutions while maintaining population diversity, hence increasing the FJSP solution efficiency. This strategy also coordinates inter-individual information exchange and effective gene recombination, mitigates premature convergence, and boosts overall population solution quality, promoting superior global and meticulous local searches.

2.3.5. Mutation Operation

This study introduced an adaptive mutation approach to enhance the algorithm's comprehensive search capability in solving FJSP and avoiding premature convergence to local optima [72]. This approach comprises two core strategies designed to optimize the overall scheduling performance [54]:

(1) Differentiated mutation probabilities are assigned to operations based on their significance:

Critical operations, such as those with longer processing times or a substantial impact on machine utilization rates, are attributed higher mutation probabilities p_{high}. In contrast, operations with less impact are assigned lower mutation probabilities p_{low}. Thus, the mutation probability p_m is determined by the relative importance of the operation.

(2) An initial mutation rate is set:

$$p_{m0} = 0.3 \tag{24}$$

(3) Adaptive alteration of the mutation rate is performed based on the change in the population's average fitness during iterations, as calculated by Formulas (5)–(21), to derive the fitness improvement indicator Δfit.

(4) The adjustment of the mutation rate follows two adaptive factors, α >1 and β <1, to increase or decrease the mutation rate, respectively. Absolute thresholds for Δfit are set at 0.01 following research by K. DeJong and others [69,70], with fitness improvements over 1% indicating significant population shifts and those under 1% representing minor changes.

The mutation rate adjustment obeys the following:

$$p_m = \begin{cases} \min(p_{mub}, p_m \cdot \alpha), \Delta fit < 0.01 \\ \max(p_{mlb}, p_m \cdot \beta), \Delta fit \geq 0.01 \end{cases} \tag{25}$$

where p_{mlb} and p_{mub} are the lower and upper bounds of the mutation rate, respectively, ensuring that the mutation rate fluctuates within a specified range. A low mutation rate may fail to introduce new potential solutions, weakening the algorithm's ability to escape local optima; conversely, a high rate may render the algorithm overly random, destroying beneficial genes. Based on Back's [73] assertion that lower mutation rates help to maintain stability while avoiding excessive randomness and Holland's [74] theory that higher rates may negatively impact convergence, preventing exploratory excess that leads to performance degradation or explosive searching, the upper limit of the mutation probability is set to 0.4 and the lower limit to 0.1. This ensures that sufficient diversity is retained, even as the population converges to evade potential local optima traps. This mutation rate range also allows the algorithm to balance exploration with exploitation, fine-tuning existing quality solutions. The mutation rate range for this study is set between 0.1 and 0.4.

The flowchart for the adaptive mutation operation is illustrated in Figure 8.

Combining the aforementioned adaptive mechanisms and differentiated mutation strategies offers FJSP an efficient optimization method. This dual adaptive design not only achieves a balance between global search and local exploration, but also demonstrates advantages in optimizing processing times and resource allocation, avoiding premature convergence, promoting extensive solution space exploration, and maintaining diversity within the population. It also enhances the algorithm's robustness at various stages and improves its applicability in real-world production environments.

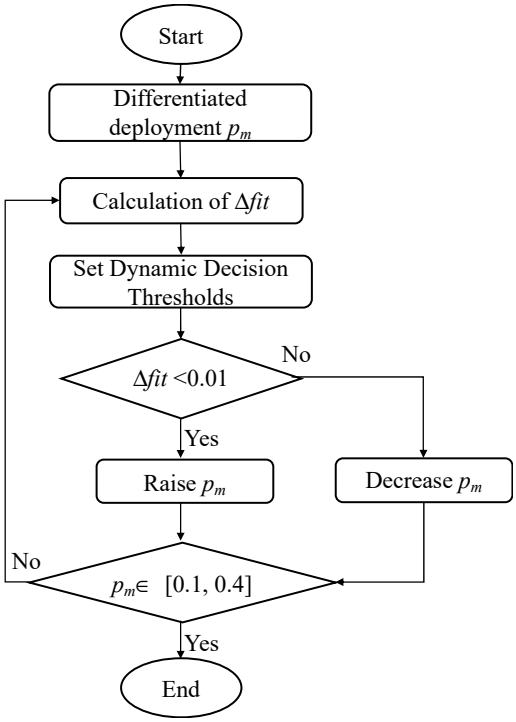

Figure 8. The flowchart of the adaptive mutation operation.

3. Results

3.1. Simulation System for Solid Wood Panel Processing

A comprehensive simulation system for solid wood panel processing was developed within Unity3D. The system allows for the input of AGV speeds via the interface to produce simulated production effects that correspond to these speeds, enhancing the visualization of the manufacturing process. Additionally, the simulation encompasses the capacity to model various manufacturing processes with differing quantities of processing equipment.

One key feature of the simulation is an interactive production scheduling Gantt chart that dynamically adjusts in real time to represent changes within the manufacturing workflow. It operates in unison with other integrated modules, such as the workshop information display and the AGV speed control module, to provide a comprehensive view of the workshop's operational status. These functionalities are pivotal in testing and validating the AIOGA performance under various scenarios.

The Unity3D simulation system, enriched with these capabilities, serves as a test bed for AIOGA and a visual aid for understanding and optimizing production logistics. Figure 9a–d illustrates how AGVs and industrial robots move according to the logic of the production line.

The simulation system also features an operator interface. When the operator interface is activated by checking the "Operator Interface Switches" button, the simulation screen displays information about the processing lines workshop, such as the set AGV speeds, production scheduling Gantt charts, operator interface switches, number of checked-in pieces, and other workshop details. When the operator interface is deactivated, the entire simulation system will only display the operational state of the simulation workshop without any data panels. The effect of the activated operator interface is shown in Figure 10. Conversely, when the "Operator Interface Switches" button is unchecked, resulting in the deactivation of the operator interface, the entire simulation system will only display the operational state of the simulation workshop without any data panels, as depicted in Figure 11.

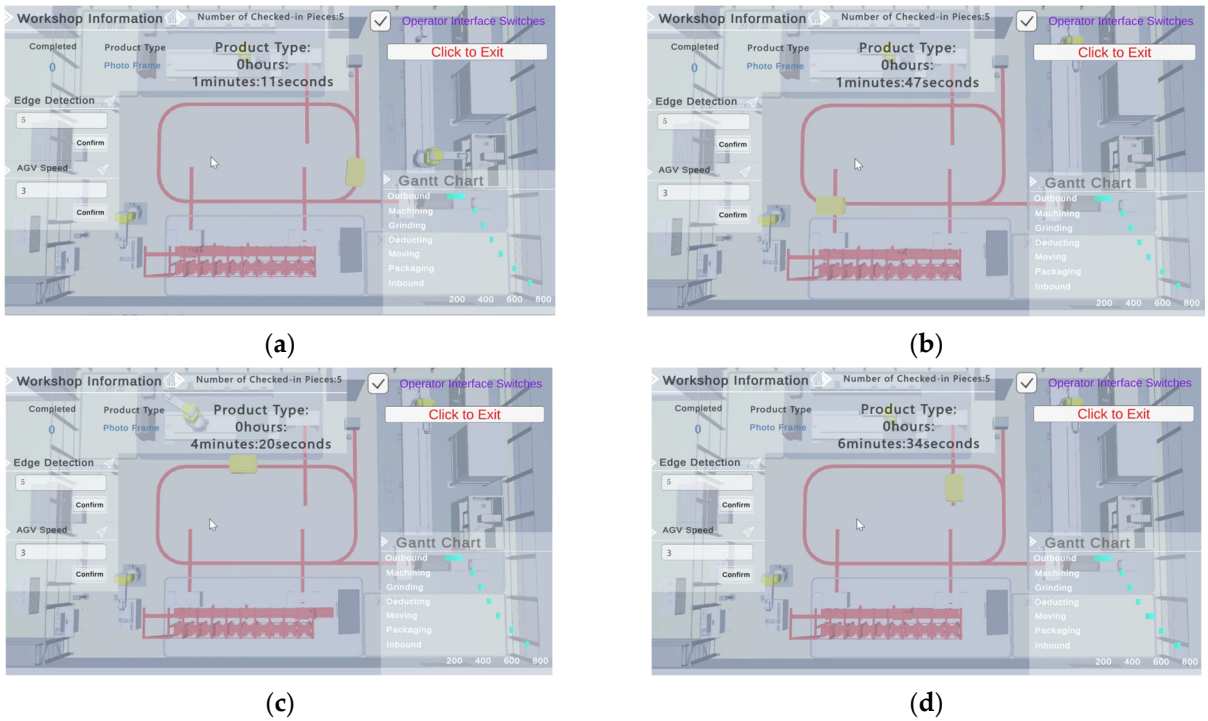

Figure 9. Unity3D simulation environment effect.

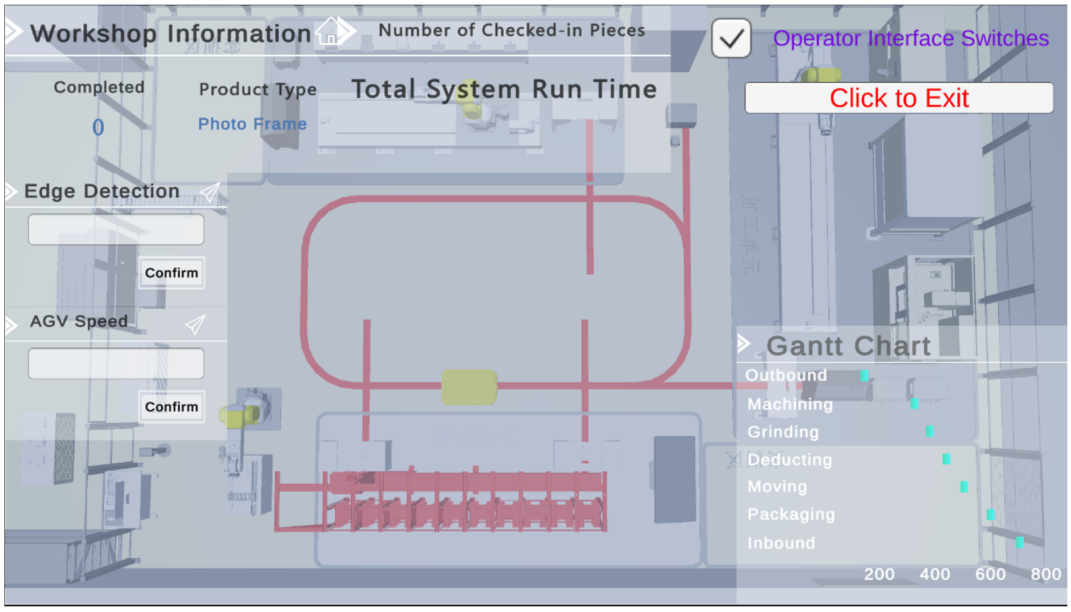

Figure 10. Simulation system interface.

In the simulation interface of the application layer, three production lines are established, and Table 4 displays the number of devices in each phase of this production line. In this study, production line 1 is configured with a distinct setup, where the outbound unit has one machine, the processing unit has five machines, the grinding unit includes four machines, dedusting is performed by one machine, and the packaging unit has two machines each. production line 2 has one, three, two, one, and one machine(s) for the outbound unit, machining, grinding, dedusting, and packaging unit. In contrast, production line 3 represents the actual flexible job-shop floor for the solid wood panel processing currently in operation, which is a single machine for each stage of the processing flow, signifying a lean production line. Production lines 1 and 2 are virtual lines construed within the simulation

system, meticulously curated to assess the computational prowess of AIOGA in addressing FJSP. These lines test the effectiveness of AIOGA under different operational circumstances and configurations, thus providing a robust examination of the algorithm's performance and flexibility in a controlled experimental environment.

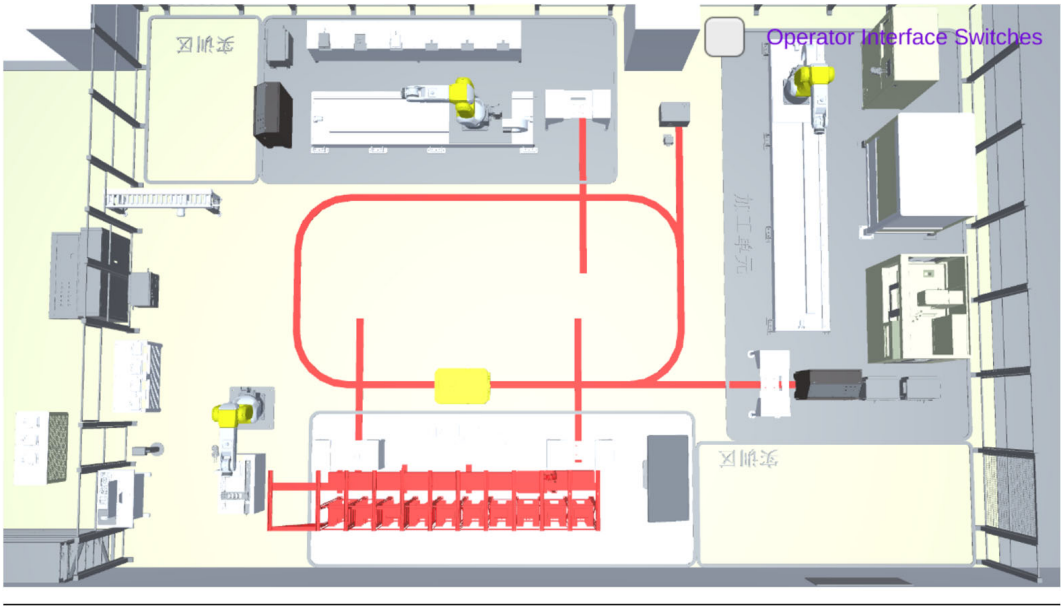

Figure 11. Operator interface closed.

Table 4. Number of process equipment.

Process	Line 1	Line 2	Line 3
Outbound	2	1	1
Machining	5	3	1
Grinding	4	2	1
Dedusting	1	1	1
Packaging	1	1	1

3.2. Application of AIOGA in Solving FJSP

This study leverages MATLAB to validate AIOGA's efficacy in solving FJSP. MATLAB was selected for its comprehensive environment and advanced optimization toolboxes, which are well-suited for simulating complex systems such as flexible job-shop production lines. Through MATLAB, we designed a simulation framework that enables the application of AIOGA and facilitates an in-depth analysis of its potential to enhance scheduling efficiency and adaptability. This selection underscores the commitment to employing robust and recognized computational tools in the research methodology, ensuring reliability and replicability in job-shop scheduling optimization.

Within this simulation, ten solid wood panels were scheduled for production. Each panel was assigned a unique identifier in the Gantt chart, corresponding to numbers 1 through 10, resulting in ten distinct numerical labels being represented within the chart. Furthermore, each panel's corresponding number was represented by a consistent color, with different colors used to distinguish the various panels. This color coding would enhance the visual differentiation for users observing the production schedule within the simulation. The production scheduling Gantt charts for these panels can be observed in Figures 12–14. Upon examining Figures 12–14, it is evident that each panel represents a specific number, and the same panel uses the same number and color for better visual distinction. This color-coding and numerical representation system enhances the clarity

of the scheduling process. Additionally, each machine can indicate which panel is being processed at any given moment. The Gantt chart facilitates the management of the manufacturing process and provides staff members on the shop floor with decision-making instructions. It enables them to lead and monitor each step effectively.

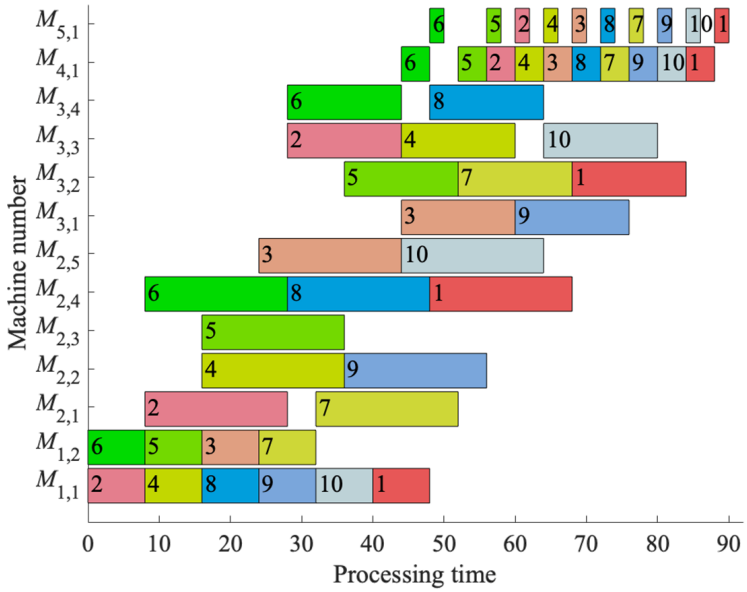

Figure 12. Gantt chart of production line 1 scheduling results.

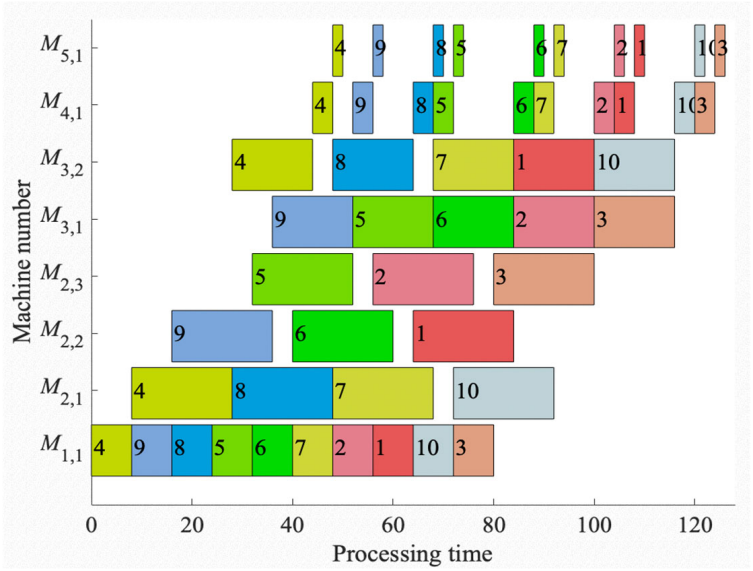

Figure 13. Gantt chart of production line 2 scheduling results.

Specifically, the production scheduling algorithm reduced the production time, as evidenced by the fact that took 90 min to produce ten products. The production of solid wood panels is feasible. According to Figures 15–17, the maximum completion time for production line 1 was 90 s, for production line 2 was 126 s, and for production line 3 was 230 s. The maximum completion time of line 1 was the smallest.

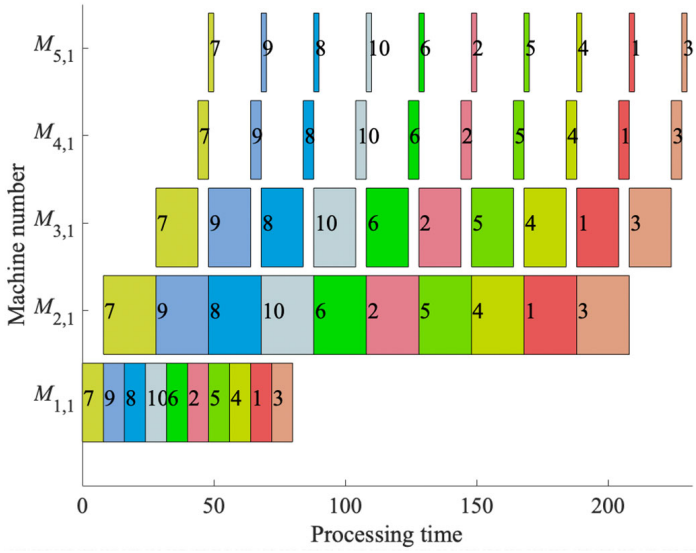

Figure 14. Gantt chart of production line 3 scheduling results.

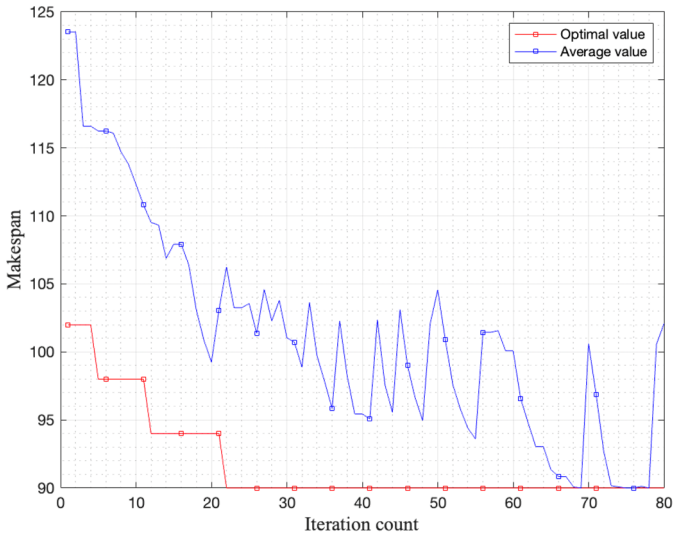

Figure 15. Production line 1 convergence comparison chart.

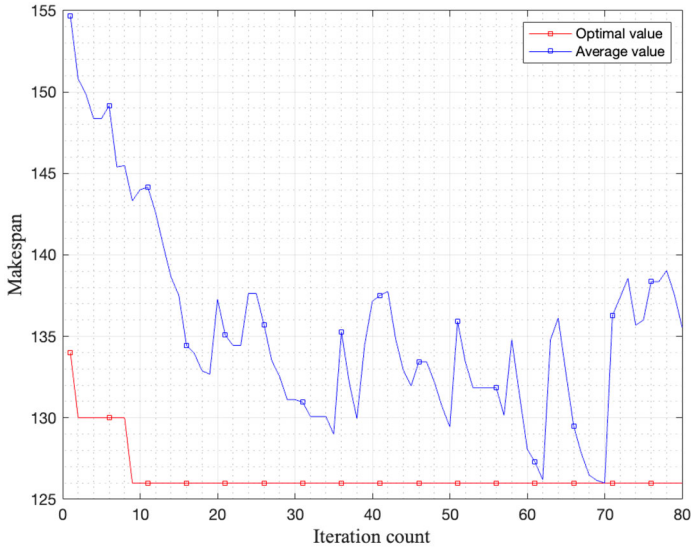

Figure 16. Production line 2 convergence comparison chart.

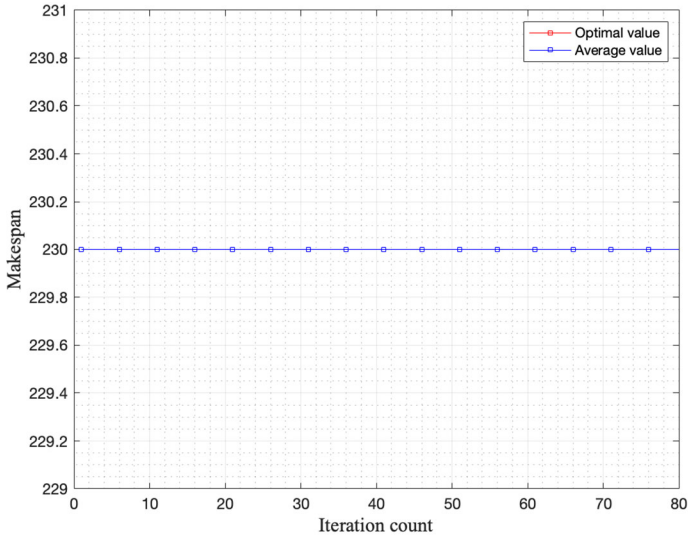

Figure 17. Production line 3 convergence comparison Chart.

4. Discussion

The performance of AIOGA was evaluated by comparing it with various heuristic and machine learning algorithms. Performance metrics such as average completion time, workload balance, resource utilization rate, and maximum completion time percentage deviation were introduced. The performances of both algorithms were compared using these measures to determine the more efficient algorithm. All algorithms utilized the number of machines and operation configuration of production line 1.

By conducting multiple algorithm runs and averaging the results, biases induced by random factors could be mitigated, making the outcomes more universally applicable and valuable. According to the existing academic literature and research practices, calculating the average over 20 runs is a common approach. This method is based on the central limit theorem, which posits that, regardless of the population data distribution, as long as the sample size is sufficiently large, the sampling distribution of the mean approximately follows a normal distribution. A cycle of 20 iterations is considered to provide adequately comprehensive sampling, making the resulting average a reliable standard for assessing algorithm performance. Therefore, this study selected the average completion time to evaluate the AIOGA's performance, adopting a strategy of 20 cycles of algorithm runs. The average completion time was the mean of the maximum completion times obtained from 20 repetitions.

Workload Balance [75] (WLB) reflects the uniformity of the task distribution across all machines. It was assessed by a statistical comparison of each machine's workload under the algorithm, serving not only as a performance measure, but also as an indicator of the production line setup's appropriateness. WLB holds critical significance for sustainable production. Firstly, it facilitates optimized resource utilization and ensures balanced equipment operation, extending the machinery's lifespan and mitigating the need for new equipment and the generation of obsolete machinery. Secondly, by minimizing equipment overuse and idle time, WLB can also reduce the waste of energy.

Therefore, implementing and refining workload balancing can enhance production efficiency, conserve resources and energy, and reduce environmental impact. This is closely aligned with sustainable production goals, underscoring the pivotal role of WLB in the pursuit of ecological and operational harmony in manufacturing processes. WLB can be calculated as follows:

$$\mathrm{WLB} = 1 - \frac{{\sigma_T}^2}{\mu_T} \tag{26}$$

where ${\sigma_T}^2$ represents the variance of all the machines' working time and μ_T represents all the machines' mean working time.

Resource Utilization Rate [76] (RUR) was introduced to measure the extent to which all available resources were utilized, reflecting the efficiency and effectiveness of the algorithm. Enhancing the RUR is a fundamental strategy for boosting the production efficiency and environmental sustainability in production management and sustainability studies.

A high RUR signifies a more rational and adequate consumption of materials and energy during production, thereby minimizing waste. This effectively reduces the strain on natural resources and diminishes the environmental impact, laying the groundwork for sustainable production practices. Enhancing the RUR can significantly reduce the generation of waste materials and by-products in the production process, decreasing the demand for waste management and mitigating related environmental pollution issues. This advancement aids businesses and society at large in progressing toward carbon footprint reduction and ecosystem conservation goals.

Thus, enhancing the RUR contributes to the sustainability of production activities, reflecting its vital role in the quest for operational efficiency and environmental stewardship in manufacturing. It can be calculated using the formula:

$$\mathrm{RUR} = \left(\frac{\sum_{i=1}^{n} W_i}{M \times T_{\max}} \right) \times 100\% \tag{27}$$

where W_i denotes the actual working time of a machine, M is the number of machines, and T_{max} represents the maximum possible working time for each machine in the production cycle.

The percentage deviation of the maximum completion time Equation (28) [77] is used as a measure:

$$P = \frac{T_x - T_0}{T_x} \times 100\% \tag{28}$$

where T_x is the maximum completion time by other methods and T_0 is the maximum completion time by the current method.

The operational results of AIOGA and other algorithms on production line 1 are analyzed from the dimensions above. Specifically, for production line 1, the maximum completion time required by AIOGA was 90. Comparing the percentage deviation of the maximum completion times, AIOGA improved by 39.60% over GA. The comparative results of the algorithms are detailed in Table 5.

Table 5. Comparison of different methods.

Algorithm Name	Best Case	Average Case	Worst Case	Workload Balance	Resource Utilization Rate	p
AIOGA	90	90	92	0.73	88.69%	0%
GA	149	151	155	0.68	82%	39.60%
ACO	95	98	100	0.72	84%	5.26%
TS	97	99	105	0.70	83%	7.22%
PSO	92	95	96	0.69	85%	2.17%
SA	109	111	120	0.74	87%	17.43%
EDA	105	106	109	0.64	80%	14.29%
HSA	100	103	105	0.67	81%	10.00%
CSA	104	106	109	0.66	80%	15.6%
ANN	86	87	89	0.75	88%	−4.65%
DL	84	85	87	0.76	90%	−7.14%
APL	230	231	240	0.31	63%	60.87%

The primary distinction between machine learning and AIOGA lies in their approaches to problem solving and the resources they require. Machine learning methods based on big data and neural networks are commonly utilized for complex pattern recognition and

predictive tasks, such as voice and image recognition. However, these methods necessitate a voluminous dataset for model training and expensive hardware resources, such as high-performance computational power and substantial storage capacity. The training process for machine learning algorithms often requires close collaboration with artificial intelligence experts, making it a less economical solution for small production enterprises with limited resources.

On the other hand, AIOGA, a form of heuristic search technique, is employed chiefly to solve optimization issues, such as our research focus, FJSP. This method does not demand extensive sample data, it merely requires the formulation of an appropriate fitness function and genetic operators (such as mutation and crossover) to search for potential optimal solutions. Hence, AIOGA is more straightforward and practical to implement, requires relatively lower hardware resources, and exhibits excellent parallelism, making it more suitable for resolving fine-grained, distributed, and dynamic problems.

At the heart of sustainable production is maximizing economic, environmental, and social benefits through heightened efficiency and effective resource utilization. By applying genetic algorithms to solve the FJSP, we can optimize production scheduling, enhance equipment utilization, and reduce material waste, which significantly promotes sustainable production for businesses, especially small- and medium-sized enterprises. Furthermore, the low cost and ease of implementation associated with genetic algorithms align with sustainable production principles, balancing economic benefits with environmental stewardship and social responsibility.

AIOGA offers an effective solution for FJSP, with a proven computational capacity and notable improvements in production scheduling. It presents a valuable approach for FJSP, enhancing production efficiency and fostering sustainable development.

In the scope of this study, AIOGA was examined across manufacturing units of varying sizes. AIOGA demonstrated a remarkable capacity for optimizing production schedules within diverse manufacturing contexts, ranging from compact job shops to expansive industrial facilities.

One of the pivotal advantages of AIOGA is its inherent flexibility and the capability to tailor heuristic rules and operations to meet the demands and limitations of various manufacturing settings. For example, within smaller manufacturing units with relatively straightforward production processes, AIOGA can quickly adjust to enhance scheduling and resource distribution, thereby boosting operational efficiency. On the other hand, for larger manufacturing units marked by more complex production procedures and more significant variations in job types and volumes, the proficiency of AIOGA in managing multi-objective optimization proves invaluable. It navigates through competing goals, such as reducing production time while optimizing resource utilization, guaranteeing efficient and sustainable production processes.

Empirical research and simulation studies further reveal that genetic algorithms, including AIOGA, can manage batch sizes and production schedules in manufacturing systems of varied scales, adeptly adapting to each one's unique set of challenges. The modular structure of AIOGA also facilitates seamless incorporation into current manufacturing systems, maintaining that its scalability does not detract from its performance.

To summarize, deploying AIOGA across manufacturing units of different scales significantly broadens its applicability and highlights its potential in promoting sustainable production methodologies throughout the manufacturing domain. The flexibility and efficiency of AIOGA in handling varied production settings denote its potential critical role in advancing manufacturing optimization, particularly within the realms of Industry 4.0 and smart manufacturing innovations.

The discussion, thus far, has focused on the practical applications and impacts of AIOGA. Following this, an exploration of its theoretical influences will be presented.

AIOGA significantly enriches the theoretical foundations of heuristic algorithms, particularly GAs. AIOGA provides a nuanced evolution of GA capabilities by introducing adaptive hyper-parameter adjustments. These adjustments enhance the exploration and

exploitation balance within the search space, enabling the algorithm to adapt dynamically to the optimization problem. This innovation elevates GA's computational strength and demonstrates the potential of intelligent adaptation in traditional heuristic frameworks.

AIOGA's adaptive nature posits an advanced, more responsive approach to parameter tuning, contributing to the broader understanding of how heuristic parameters can be self-regulated in response to varying problem complexities. This advance can inform future research into heuristic algorithm design and lead to the development of more sophisticated, self-adjusting algorithms that maintain a high degree of solution quality across diverse problem sets.

In summary, AIOGA represents a stride in the theory of GAs, exhibiting how strategic hyper-parameter adaptation can reinforce core algorithmic efficiency. Its practical ability to solve FJSP is a testament to the adaptability afforded by such theoretical advancements, which mark the evolution in the field of heuristic algorithms.

5. Conclusions

In the realm of sustainable manufacturing, efficient resource utilization and the minimization of wastage are fundamental pillars. This research contributes to these principles by focusing on the sustainable production of solid wood panels, an area ripe for advancements in operational optimization. By streamlining processes through intelligent scheduling, the environmental impact of manufacturing can be significantly reduced without compromising productivity.

The research delves into constructing a simulation system for solid wood panel processing lines and applying the Adaptive Intelligent Optimization Genetic Algorithm (AIOGA) to solve the Flexible Job-Shop Scheduling Problem (FJSP).

Initially, the Unity3D platform delineated the processing production logic in constructing the solid wood panel simulation system. The interface displayed simulation models of the production line, where the number of machines and the AGV speed could be adjusted, allowing the simulation results to be attained using subsequently developed algorithms. In terms of the algorithmic model construction, after studying various algorithms, GA was enhanced and AIOGA was constructed. Within AIOGA, a hybrid coding strategy effectively transformed the production scheduling problem into a genetic operation basis, enabling the assessment of each solution's quality according to the objective function, which focused on minimizing the overall production completion time. The strategy also dynamically selected the appropriate approach according to real-life situations.

In AIOGA, crossover and mutation operations were likewise adaptive. The dynamic crossover operation started from a baseline value of 0.8. It adjusted its rate in real time to enhance the diversity of solutions and cover the search space, ensuring the crossover rate control range was between 0.6 and 0.9. The adaptive mutation operation aimed to increase genetic diversity and prevent the premature convergence of the algorithm. With a mutation rate set at 0.3, it adaptively adjusted from 0.1 to 0.4. AIOGA flexibly adjusted its key parameters and offered practical solutions to intelligent scheduling issues unique to solid wood panel processing.

Comparisons with the Genetic Algorithm (GA) indicated a maximum completion time percentage deviation of 39.60% and a maximum completion time of 90 min, demonstrating that AIOGA exhibited robustness and adaptability in dealing with FJSP. It enhanced the efficiency and accuracy of production scheduling.

Author Contributions: Conceptualization, J.Y.; methodology, J.Y.; software, J.Y.; validation, J.Y.; formal analysis, J.Y.; investigation, J.Y.; resources, Y.Z. and J.W.; data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, J.W.; visualization, J.W.; supervision, Y.Z. and J.W.; project administration, J.W.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beijing Forestry University Excellent Experimenter Item, grant number BJFUSY20220707.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. *Ecosystems and Human Well-Being: Synthesis*; Millennium Ecosystem Assessment (Program) (Ed.); Island Press: Washington, DC, USA, 2005; ISBN 978-1-59726-040-4.
2. Vilar-Dias, J.L.; Junior, A.S.S.; Lima-Neto, F.B. An Interpretable Digital Twin for Self-Aware Industrial Machines. *Sensors* **2023**, *24*, 4. [CrossRef]
3. Nahavandi, S. Industry 5.0—A Human-Centric Solution. *Sustainability* **2019**, *11*, 4371. [CrossRef]
4. Bonello, A.; Francalanza, E.; Refalo, P. Smart and Sustainable Human-Centred Workstations for Operators with Disability in the Age of Industry 5.0: A Systematic Review. *Sustainability* **2023**, *16*, 281. [CrossRef]
5. Antov, P.; Lee, S.H.; Lubis, M.A.R.; Kristak, L.; Réh, R. Advanced Eco-Friendly Wood-Based Composites II. *Forests* **2023**, *14*, 826. [CrossRef]
6. Ševčíková, R.; Knošková, Ľ. Sustainable Design in the Furniture Industry. In Proceedings of the 21st International Joint Conference Central and Eastern Europe in the Changing Business Environment: Proceedings, Prague, Czech Republic, 20–21 May 2021.
7. Wang, L.; He, J.; Xu, S. The Application of Industry 4.0 in Customized Furniture Manufacturing Industry. *MATEC Web Conf.* **2017**, *100*, 03022. [CrossRef]
8. Awouda, A.; Traini, E.; Bruno, G.; Chiabert, P. IoT-Based Framework for Digital Twins in the Industry 5.0 Era. *Sensors* **2024**, *24*, 594. [CrossRef]
9. Merkuryeva, G.; Shires, N. Manufacturing System Planning and Scheduling. In *Simulation-Based Case Studies in Logistics*; Merkuryev, Y., Merkuryeva, G., Piera, M.À., Guasch, A., Eds.; Springer: London, UK, 2009; pp. 19–33, ISBN 978-1-84882-186-6.
10. Yang, L.; Li, J.; Hackney, P.; Chao, F.; Flanagan, M. Manual Task Completion Time Estimation for Job Shop Scheduling Using a Fuzzy Inference System. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 139–146.
11. Holubek, R.; Kostal, P. The Intelligent Manufacturing Systems. *Adv. Sci. Lett.* **2013**, *19*, 972–975. [CrossRef]
12. Mittal, S.; Khan, M.A.; Romero, D.; Wuest, T. Smart Manufacturing: Characteristics, Technologies and Enabling Factors. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2019**, *233*, 1342–1361. [CrossRef]
13. Shang, X. A Study of Deep Learning Neural Network Algorithms and Genetic Algorithms for FJSP. *J. Appl. Math.* **2023**, *2023*, 4573352. [CrossRef]
14. Buddala, R.; Mahapatra, S.S.; Singh, M.R.; Balusa, B.C.; Balam, V.P. Improved TLBO and JAYA Algorithms to Solve New Fuzzy Flexible Job-Shop Scheduling Problems. *J. Ind. Eng. Int.* **2022**, *18*, 102–114.
15. Reijnen, R.; van Straaten, K.; Bukhsh, Z.; Zhang, Y. Job Shop Scheduling Benchmark: Environments and Instances for Learning and Non-Learning Methods. *arXiv* **2023**, arXiv:2308.12794.
16. Dauzère-Pérès, S.; Ding, J.; Shen, L.; Tamssaouet, K. The Flexible Job Shop Scheduling Problem: A Review. *Eur. J. Oper. Res.* **2024**, *314*, 409–432. [CrossRef]
17. Grieves, M. Deep Reinforcement Learning for Inventory Optimization with Non-Stationary Uncertain Demand. *Digit. Twin White Pap.* **2014**, 1–7.
18. Dong, W.; Jin, M. Automated Storage and Retrieval System Design with Variant Lane Depths. *Eur. J. Oper. Res.* **2024**, *314*, 433–445. [CrossRef]
19. Omara, F.A.; Arafa, M.M. Genetic Algorithms for Task Scheduling Problem. *J. Parallel Distrib. Comput.* **2010**, *70*, 13–22. [CrossRef]
20. Eglese, R.W. Simulated Annealing: A Tool for Operational Research. *Eur. J. Oper. Res.* **1990**, *46*, 271–281. [CrossRef]
21. Glover, F. Tabu Search: A Tutorial. *Interfaces* **1990**, *20*, 74–94. [CrossRef]
22. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
23. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [CrossRef]
24. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
25. Glover, F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Comput. Oper. Res.* **1986**, *13*, 533–549. [CrossRef]
26. Kennedy, J. Swarm Intelligence. In *Handbook of Nature-Inspired and Innovative Computing*; Zomaya, A.Y., Ed.; Kluwer Academic Publishers: Boston, MA, USA, 2006; pp. 187–219, ISBN 978-0-387-40532-2.
27. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milan, Italy, 1992.
28. Hopfield, J.J.; Tank, D.W. "Neural" Computation of Decisions in Optimization Problems. *Biol. Cybern.* **1985**, *52*, 141–152. [CrossRef] [PubMed]

29. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
30. Goldberg, D.E.; Lingle, R. Alleles, Loci, and the Traveling Salesman Problem. In Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburg, PA, 24–26 July 1985; Psychology Press: London, UK, 1985, ISBN 978-1-315-79967-4.
31. Zhang, G.; Gao, L.; Shi, Y. An Effective Genetic Algorithm for the Flexible Job-Shop Scheduling Problem. *Expert Syst. Appl.* **2011**, *38*, 3563–3573. [CrossRef]
32. Roeva, O.; Fidanova, S.; Paprzycki, M. Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modelling. *Fed. Conf. Comput. Sci. Inf. Syst.* **2013**, *2013*, 371–376.
33. Fogel, D.B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*; John Wiley & Sons: Hoboken, NJ, USA, 2006. Available online: https://ieeexplore.ieee.org/book/5237910 (accessed on 15 April 2024).
34. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998. Available online: https://direct.mit.edu/books/book/4675/An-Introduction-to-Genetic-Algorithms (accessed on 5 April 2024).
35. Grefenstette, J.J. Optimization of Control Parameters for Genetic Algorithms. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 122–128. Available online: https://ieeexplore.ieee.org/document/4075583 (accessed on 5 April 2024). [CrossRef]
36. Baker, J.E. Reducing Bias and Inefficiency in the Selection Algorithm. In Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application, Cambridge, MA, USA, 28–31 July 1987. Available online: https://dl.acm.org/doi/10.5555/42512.42515 (accessed on 5 April 2024).
37. Whitley, L.D. A Genetic Algorithm Tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [CrossRef]
38. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley Pub. Co.: Boston, MA, USA, 1989. [CrossRef]
39. Back, T. Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 27–29 June 1994. Available online: https://ieeexplore.ieee.org/document/350042 (accessed on 5 April 2024).
40. Baker, J.E. Adaptive Selection Methods for Genetic Algorithms. In Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, USA, 24–24 July 1985.
41. Goldberg, D.E.; Deb, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms*; Elsevier: Amsterdam, The Netherlands, 1991; Volume 1, pp. 69–93, ISBN 978-0-08-050684-5.
42. Kumar, A.; Liu, B.; Miikkulainen, R.; Stone, P. Effective Mutation Rate Adaptation through Group Elite Selection. In Proceedings of the Genetic and Evolutionary Computation Conference, Boston, MA, USA, 9–13 July 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 721–729.
43. Srinivas, M.; Patnaik, L.M. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Trans. Syst. Man Cybernetics* **1994**, *24*, 656–667. Available online: https://ieeexplore.ieee.org/document/286385 (accessed on 4 April 2024). [CrossRef]
44. Sivanandam, S.N.; Deepa, S.N. Introduction to Genetic Algorithms. 2023, 6. Available online: https://link.springer.com/book/10.1007/978-3-540-73190-0 (accessed on 6 April 2024).
45. Haupt, R.L.; Haupt, S.E. *Practical Genetic Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
46. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*; Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.-P., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 849–858.
47. Konak, A.; Coit, D.W.; Smith, A.E. Multi-Objective Optimization Using Genetic Algorithms: A Tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007. [CrossRef]
48. Bandaru, S.; Tulshyan, R.; Deb, K. Modified SBX and Adaptive Mutation for Real World Single Objective Optimization. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1335–1342.
49. Hinterding, R.; Michalewicz, Z.; Eiben, A.E. Adaptation in Evolutionary Computation: A Survey. In Proceedings of the Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), Indianapolis, IN, USA, 13–16 April 1997.
50. Eiben, A.E.; Hinterding, R.; Michalewicz, Z. Parameter Control in Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **1999**, *3*, 124–141. [CrossRef]
51. Semenkin, E.; Semenkina, M. Self-Configuring Genetic Programming Algorithm with Modified Uniform Crossover. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–6.
52. Watanabe, M.; Ida, K.; Gen, M. A Genetic Algorithm with Modified Crossover Operator and Search Area Adaptation for the Job-Shop Scheduling Problem. *Comput. Ind. Eng.* **2005**, *48*, 743–752. [CrossRef]
53. Viana, M.S.; Morandin Junior, O.; Contreras, R.C. A Modified Genetic Algorithm with Local Search Strategies and Multi-Crossover Operator for Job Shop Scheduling Problem. *Sensors* **2020**, *20*, 5440. [CrossRef] [PubMed]
54. Wang, L.; Zheng, D.Z. An Effective Hybrid Optimization Strategy for Job-Shop Scheduling Problems. *Comput. Oper. Res.* **2001**, *28*, 585–596. [CrossRef]
55. Fogel, D.B. *Artificial Intelligence through Simulated Evolution*; Wiley-IEEE Press: Hoboken, NJ, USA, 1998. Available online: https://www.semanticscholar.org/paper/Artificial-Intelligence-through-Simulated-Evolution-Fogel-Owens/69022c885504a091680cf2dc9cfc84597332ac69 (accessed on 6 April 2024).
56. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer: Berlin/Heidelberg, Germany, 2024.

57. Fast Evolution Strategies. Available online: https://www.researchgate.net/publication/225205181_Fast_Evolution_Strategies (accessed on 6 April 2024).
58. Matousek, R.; Nolle, L. GAHC: Improved GA with HC Mutation. *Lect. Notes Eng. Comput. Sci.* **2007**, *2167*, 24–26.
59. Rajakumar, B. Impact of Static and Adaptive Mutation Techniques on the Performance of Genetic Algorithm. *Int. J. Hybrid Intell. Syst.* **2013**, *10*, 11–22. [CrossRef]
60. Khair, U.; Lestari, Y.D.; Perdana, A.; Hidayat, D.; Budiman, A. Genetic Algorithm Modification Analysis of Mutation Operators in Max One Problem. In Proceedings of the 2018 Third International Conference on Informatics and Computing (ICIC), Palembang, Indonesia, 17–18 October 2018. [CrossRef]
61. Neubauer, A. A Theoretical Analysis of the Non-Uniform Mutation Operator for the Modified Genetic Algorithm. In Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), Indianapolis, IN, USA, 13–16 April 1997; pp. 93–96.
62. Synergistic Effect of Well-Defined Dual Sites Boosting the Oxygen Reduction Reaction—Energy & Environmental Science (RSC Publishing). Available online: https://pubs.rsc.org/en/content/articlelanding/2018/ee/c8ee02656d (accessed on 4 April 2024).
63. Allahverdi, A. The Third Comprehensive Survey on Scheduling Problems with Setup Times/Costs. *Eur. J. Oper. Res.* **2015**, *246*, 345–378. [CrossRef]
64. Scheduling: Theory, Algorithms, and Systems | SpringerLink. Available online: https://link.springer.com/book/10.1007/978-3-319-26580-3 (accessed on 4 April 2024).
65. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
66. Back, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*; Oxford University Press: Oxford, UK, 1996.
67. Blickle, T.; Thiele, L. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evol. Comput.* **1996**, *4*, 361–394. [CrossRef]
68. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Available online: https://www.semanticscholar.org/paper/SPEA2:-Improving-the-strength-pareto-evolutionary-Zitzler-Laumanns/b13724cb54ae4171916f3f969d304b9e9752a57f (accessed on 5 April 2024).
69. DeJong, K. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Thesis, University of Michigan: Ann Arbor, MI, USA, 1975.
70. Jia, Y.; Yang, X. Optimization of Control Parameters Based on Genetic Algorithms for Spacecraft Attitude Tracking with Input Constraints. *Neurocomputing* **2016**, *177*, 334–341. [CrossRef]
71. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2015. Available online: https://link.springer.com/book/10.1007/978-3-662-44874-8 (accessed on 14 April 2024).
72. Wang, Y.M.; Yin, H.L.; Qin, K.D. A Novel Genetic Algorithm for Flexible Job Shop Scheduling Problems with Machine Disruptions. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 1317–1326. Available online: https://link.springer.com/article/10.1007/s00170-013-4923-z (accessed on 4 April 2024). [CrossRef]
73. Bäck, T. Optimal Mutation Rates in Genetic Search. In Proceedings of the 5th International Conference on Genetic Algorithms, Champaign, IL, USA, 17–23 July 1993; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993; pp. 2–8.
74. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992. Available online: https://ieeexplore.ieee.org/book/6267401 (accessed on 14 April 2024).
75. Kurz, M.E.; Askin, R.G. Comparing Scheduling Rules for Flexible Flow Lines. *Int. J. Prod. Econ.* **2003**, *85*, 371–388. [CrossRef]
76. Antony, J.; Kumar, M.; Labib, A. Gearing Six Sigma into UK Manufacturing SMEs: Results from a Pilot Study. *J. Oper. Res. Soc.* **2008**, *59*, 482–493. Available online: https://www.semanticscholar.org/paper/Gearing-Six-Sigma-into-UK-manufacturing-SMEs:-from-Antony-Kumar/670a02596138ab6ec149f098dfa2762697ff7da8 (accessed on 12 April 2024). [CrossRef]
77. Wang, J.X. Mixed flow shop scheduling solution for panel furniture with buffer constraints. *J. For. Eng.* **2023**, *3*, 198–204.