machines

Article

Cyber-Physical Scheduling System for Multiobjective Scheduling Optimization of a Suspension Chain Workshop Using the Improved Non-Dominated Sorting Genetic Algorithm II

Wenbin Zhao, Junhan Hu, Jiansha Lu * and Wenzhu Zhang

College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China; wenbin86@zjut.edu.cn (W.Z.); 202005031108@zjut.edu.cn (J.H.); wzzhang@zjut.edu.cn (W.Z.)
* Correspondence: ljs@zjut.edu.cn; Tel.: +86-571-88320433

Abstract: Cyber-Physical Systems (CPSs) offer significant potential to address the evolving demands of industrial development. In the Industry 4.0 era, a framework integrating sensing, data exchange, numerical analysis, and real-time feedback is essential for meeting modern industrial needs. However, implementing this integration presents challenges across multiple domains, particularly in digital analysis, information sensing, and data exchange during corporate transformation. Companies yet to undergo transformation face distinct challenges, including the risks and trial-and-error costs of adopting new technologies. This study focuses on a heavy-duty workpiece processing factory, with a specific emphasis on the painting process. The complexity of this process frequently results in congestion, which is approached as a multi-objective, multi-constraint optimization problem. This paper proposes the Improved Non-dominated Sorting Genetic Algorithm II (INSGA-II) to address the requirements of multi-objective optimization. The proposed approach uses multi-chromosome structures, listeners, and recursive backtracking initialization to optimize the search for solutions under constraints. This enables the factory to automatically streamline production lines based on workpiece processing sequences, leading to increased efficiency. Additionally, a CPS framework focused on simulation modeling has been designed. First, the INSGA-II algorithm processes order data to generate production schedules. Next, the data transmission formats and input-output interfaces are designed. Then, a simulation model is built using the algorithm's results. These components collectively form the CPS framework for this study. The proposed method offers an automated digital solution through the algorithm, enabling verification of its feasibility via the simulation model. As a result, it significantly enhances decision-making speed, reliability, and equipment utilization.

Keywords: cyber-physical system; modeling and simulation; INSGA-II; multiobjective optimization; production scheduling

check for updates

Citation: Zhao, W.; Hu, J.; Lu, J.; Zhang, W. Cyber-Physical Scheduling System for Multiobjective Scheduling Optimization of a Suspension Chain Workshop Using the Improved Non-Dominated Sorting Genetic Algorithm II. *Machines* **2024**, *12*, 666. https://doi.org/10.3390/machines12090666

Academic Editor: Kai Cheng

Received: 16 July 2024
Revised: 3 September 2024
Accepted: 16 September 2024
Published: 23 September 2024

1. Introduction

Currently, many companies are striving to transition to multivariety and small-batch production, a trend particularly prevalent in China. However, due to facility changes and production downtime, this transition often incurs high costs. In line with Industry 4.0, numerous related studies have introduced the concepts of Cyber-Physical Systems (CPSs) and digital twins. Dafflon et al. conducted a comprehensive review of extensive research on CPS [1]. From this review, they identified the challenges, methods, and technologies that drive CPS in Industry 4.0 manufacturing. Zamfirescu et al., in their research, highlighted that traditional CPS frameworks often overlook the human factor. As a result, they proposed a human-centered CPS framework to overcome this limitation [2]. Similarly, Hermann proposed principles for achieving this goal from a Six Sigma perspective [3]. These methods serve as one of the foundational elements for the design presented in this study.

In this study, we adopt a suspension-chain-production system of a particular factory as a model. The goal is to incorporate algorithms into the existing system to achieve process improvements while retaining the current hardware. Most existing studies require hardware modifications [4–6], a cost we aim to avoid. Our approach leverages existing hardware facilities considerably while constructing a CPS tailored to the current production process.

Furthermore, this study explores a multiobjective, multiconstraint scheduling problem. Notably, there is substantial research in this area. Issa categorized such problems into four types: A, B, C, and D [7], and further outlined their frameworks, as shown in Figure 1. Our study focuses on a multiobjective problem under limited resource constraints, specifically a priority-constrained resource-constrained project scheduling problem (P-RCPSP), which falls under category B. Heuristic algorithms are commonly employed for such problems, and deliver outstanding performance. For example, the study by Kai Guo et al. focuses on a cloud manufacturing service composition optimization (SCO) model [8]; Y. Wang et al. proposed an optimal combination scheme for shared manufacturing services [9]. Furthermore, Weimin Jing et al. introduced a cloud-edge collaborative scheduling mechanism aimed at optimizing the scheduling of local tasks about the cloud edge [10].

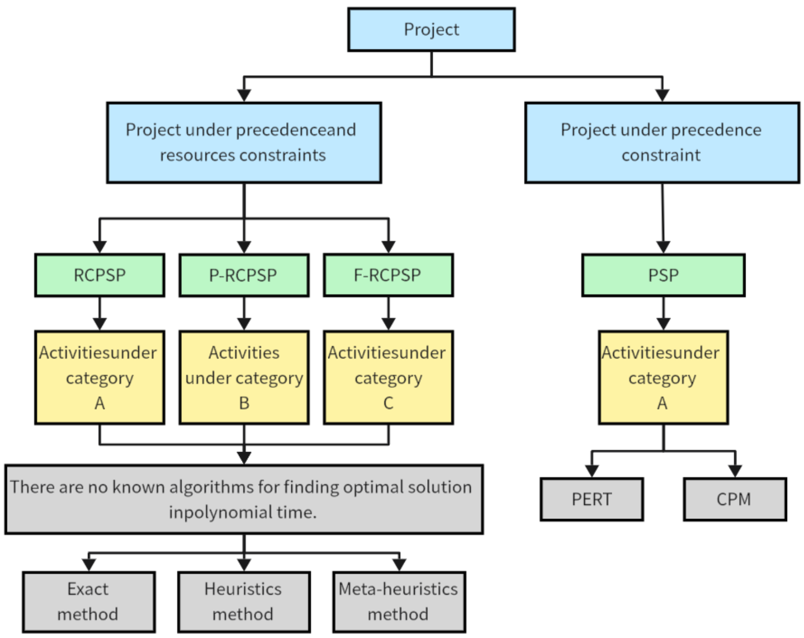

Figure 1. Classification diagram; the classifications of the project and multi-project scheduling problems.

The challenges of this problem are as follows. First, multiple constraints must be considered during the process to ensure the feasibility of the processing sequence. Additionally, the factors involved in this process include changeover times, logistics transportation times, painting time uncertainties, and workpiece reprocessing. Additionally, balancing the granularity of simulations and simulation resources is a significant challenge.

Studies that have specifically addressed the reprocessing of workpieces are few, with most focusing on designing algorithms that tackle constraints. The practical applications of such scenarios are even rarer.

The following tasks will be executed in this study. (1) Analyze the problem and define the objectives and boundary conditions or constraints. (2) Design a Cyber-Physical Scheduling System (CPSS) specifically for logistics scheduling based on existing facilities and assets. (3) Fit the parameters of the factory under actual conditions. (4) Analyze the data transmission involved in the current system, and design a scheduling model based on this analysis. (5) Design corresponding control and monitoring mechanisms around the scheduling model. (6) Integrate these mechanisms and the scheduling model into a simulation model for validation.

2. Research Background

2.1. Cyber-Physical System and Digital Twins

As production methods evolve, traditional extensive production approaches can no longer meet competitive demands. Achieving precise control is essential for significantly enhancing production capacity, which is difficult to achieve manually. This is why CPSs were introduced. Real-world control is often associated with errors that can prevent precise decision-making. Fortunately, digitization can mitigate this. With the deepening integration of digitization, intelligent algorithms are increasingly being utilized in production. This is inevitable considering the current emphasis on precision in production. However, all of these advancements require a robust real-time support system.

A CPS is a control system that integrates information acquisition, data storage, data analysis, and feedback mechanisms into a unified framework. In 2008, Sha proposed the potential of CPS development in the 21st century [11], emphasizing the importance of reliability and predictability.

As Fei-Yue Wang stated, with the advancement of intelligent systems, CPSs are widely applied in large-scale production processes [12]. Jay Lee further proposed the concept of industrial metaverse in his study [13]. They are defined as "collaborative computing system closely related to the surrounding physical world and its ongoing processes, while providing and using data access and data-processing services available on the internet" [14]. Hung-An Kao, in his research, designed an algorithm using an adaptive health monitoring algorithm through the Cyber Physical Interface for Automation Systems [15]. Meanwhile, Eyup Cinar proposed a predictive maintenance plan that integrates machine learning into CPS, achieving great results [16]. Tianjie Fu et al. designed a full-process digital twin system to visualize the converter steelmaking process, enabling better control over the process and quality. This system significantly reduced waste during smelting and nearly doubled production capacity [17].

Hoffmann et al. proposed a standardized approach to building CPSs [18]. From a Six Sigma perspective, Hermann provided a set of principles for designing CPSs from a global viewpoint [3]. Hoffmann noted that in traditional production processes, the CPS components are developed separately, and the CPS needs to integrate all instances, which is a major challenge. Therefore, in this study, we also design corresponding input/output (I/O) and data structures among the mentioned components. Data structures will be a key focus.

Regarding the maturity of CPSs, Monostori provided an evaluation standard, categorizing maturity into five levels, with the highest level being autonomous optimization [14]. Following lean thinking, this study aims to accommodate optimization, which is crucial for subsequent designs.

In addition to the methodologies and practical applications of CPSs mentioned above, several extensive studies focusing on information security have been conducted. A. Yaacoub et al. provided a comprehensive review of the potential technical standards for CPS, as well as the associated risks such as security vulnerabilities [19]. Ashraf Tantawy et al. further developed potential attack models and designed corresponding defense mechanisms, which are used to provide a solution [20]. S. Bernardi et al. proposed an anti-interference method for hyper-realistic environments, and validated it through CPS [21]. Yu proposed a comprehensive security approach across physical, virtual, and information interaction layers [22].

2.2. Multiobjective Scheduling Problem

Since Nagar et al. published a survey paper on multiobjective and bi-objective scheduling in 1995, multiobjective scheduling and flexible scheduling have gradually attracted interest across various fields [23].

Most studies on solving multiobjective problems use heuristic algorithms, such as the Genetic Algorithm (GA) [24–26], and other evolutionary algorithms, such as Grey Wolf Optimization [27], Particle Swarm Optimization (PSO) [28,29], and Bacterial Foraging

Optimization [30]. Hybrid algorithms, such as GASA, which combine GA and Simulated Annealing, have also been developed for production scheduling systems [31]. Many of these problems focus on energy optimization [32,33]. Additionally, technologies such as reinforcement learning and deep learning have also been applied to multi-objective optimization. For example, Tianjie Fu introduced the Slab Defect Generation (SDG) network, which is highly effective for image generation and defect recognition [34].

Among these, Non-dominated Sorting Genetic Algorithm II (NSGA-II) has been widely applied since its introduction in 2002 [24]. NSGA-II combines non-dominated sorting and an elitism strategy, which helps prevent premature convergence and local-optima trapping, making it advantageous for solving multidimensional multiobjective problems [35]. Similarly, other algorithms construct Pareto fronts, such as PSO [36]. However, GA provides comparatively good flexibility for different operators, which influences subsequent constraint adaptation. Additionally, NSGA-III can handle complex objective functions [37], which is crucial for practical applications in this study.

Notably, some algorithms are applied to cloud platforms, requiring a CPS architecture for real-time data support. Guo combined NSGA-II with cloud manufacturing to improve a service composition optimization model [8], making it fully mappable to physical entities. Due to the lack of flexibility in traditional cloud manufacturing service composition and scheduling, Jing proposed cloud-edge collaborative composition and scheduling [10].

2.3. Multiconstraint Scheduling Problem

In addition to solutions for multiobjective problems, complex constraint conditions are attracting significant research interest.

Fonseca proposed that the essence of evolutionary algorithms is unconstrained search [38]. Accordingly, a constrained evolutionary algorithm was introduced. From the perspective of algorithm structure, related studies on multiconstraint conditions have gradually emerged. Efrén Mezura-Montes et al. presented a prominent nature-inspired algorithms currently used for constrained numerical optimization problems (CNOP) [39]. Mengjun Ming et al. also designed an algorithm specifically for CNOP that better balances the solutions along the Pareto front under different objectives [40]. In the industrial field, this problem has become a focal point for researchers. Hongbin Qin et al. designed a discrete combinatorial optimization problem under special constraints, focusing on processing interval constraint (PIC) and job transportation time (JTT) [41]. Franco M. Novara et al. proposed a solution of large-scale scheduling problem for multi-product, multi-stage batches under constrained conditions [42]. These studies address constraint handling through direct coupling with operators.

Kui et al. considered the workpiece transportation time. Their research included adjusting operators and using inversion, crossover, and swapping for neighborhood search, which can be considered a feasible solution transformation [43]. Another approach integrated non-dominated sorting with PSO [44], one of the most common resource constraints in industrial production. These methods are similar to those involved in NSGA-II, and have shown good results when addressing multiobjective and multiconstraint problems.

Due to the limited logistics intensity of suspension chain transportation, similar to other heavy workpiece transportation methods, logistics becomes an important factor to consider during scheduling. Yiran Dan et al. proposed a process-optimization-scheduling method considering process blockage [45], encountering similar constraints, such as non-buffer areas and temporary stack storage difficulties, which are relevant to the current study. However, their situation differs from the batch production of heavy components. NSGA-II has also shown good effectiveness in solving complex constraint problems [46].

2.4. About Simulation and Modeling

Verification, Validation, and Accreditation (VV&A) is a crucial aspect of simulations, significantly impacting their credibility and reliability.

Simulation and modeling play a significant role in CPSs [47]. Through simulation modeling, one can verify the processor's results and gradually discover and improve the solution during the simulation-development process. Therefore, simulations can considerably reduce trial-and-error costs for enterprises. Regarding Modeling and Simulation (M&S), this technology is already mature abroad, with extensive commercial, industrial, and military activities revolving around M&S. However, academic researchers might be more concerned with the reliability assessment of simulation models. As M&S technology continues to develop and become widely applied, three major systems have gradually evolved for the reliability assessment of simulation models: DOD VV&A RPG 2.5, primarily developed by the Department of Defense [48]; IEEE STD 1730.2, by the Institute of Electrical and Electronics Engineers (IEEE) [49]; and SISO GMVV, by the European Committee for Standardization (CEN) [50,51].

Among these, DOD VV&A RPG 2.5 is the most prominent, and many concepts in the other two systems are derived from it, making it the earliest standard developed for VV&A research.

3. Model Establishment

This section primarily discusses the strategy for addressing current issues in planning, as well as the applicable frameworks and technologies.

3.1. Design of Cyber-Physical Scheduling Systems

The CPSS was developed based on the principles of CPS and software design. Figure 2 illustrates the design of the CPSS, drawing inspiration from classical three-tier architectures and the ModelView-Controller (MVC) principle to effectively tackle challenges.

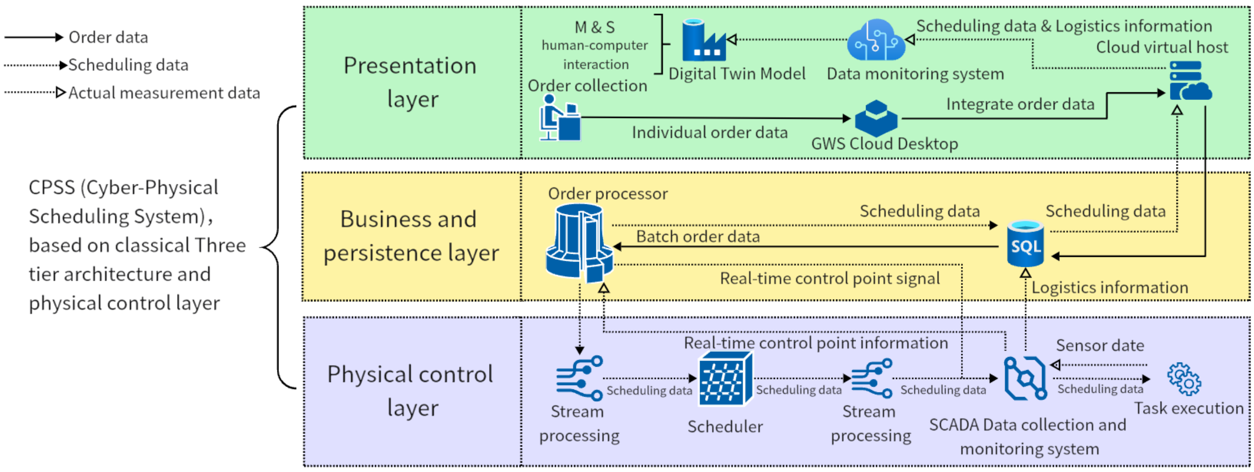

Figure 2. CPSS architecture diagram: design integrating MVC and the classical three-tier architecture to address current issues

In this context, the focus lies primarily on the order processor in the business and persistence layers. The emphasis is on designing algorithms under specific constraints and integrating them into practical applications. Assuming other layers are in place, existing facilities retain I/O interfaces. The details of these I/O interfaces will be discussed in subsequent sections.

In Figure 2, data are manually inputted from GWS cloud desktops into a cloud virtual host. The system aggregates discrete independent orders and outputs them to the database. Subsequently, these order details are processed by the order processor to derive the final scheduling results. Thereafter, these results are batched and sent by the scheduler to the data collection and monitoring system, which ultimately controls the transportation system.

Following this, real-time operational data from the system feeds back into the order processor. The order processor utilizes this real-time data to send precise control signals, facilitating real-time and accurate scheduling of operations within the suspension–chain–transportation system.

The evolution of the data-transmission structure within the system is illustrated in Figure 3. Data transmission occurs in three stages: encoding, decoding, and control. Initially, starting with the variable "orders", order data are encapsulated into a list. Subsequently, it is compiled into a two-dimensional collection containing multiple products. Each row represents components destined for respective production lines, indexed by their component numbers. These indices are crucial for chromosome encoding.

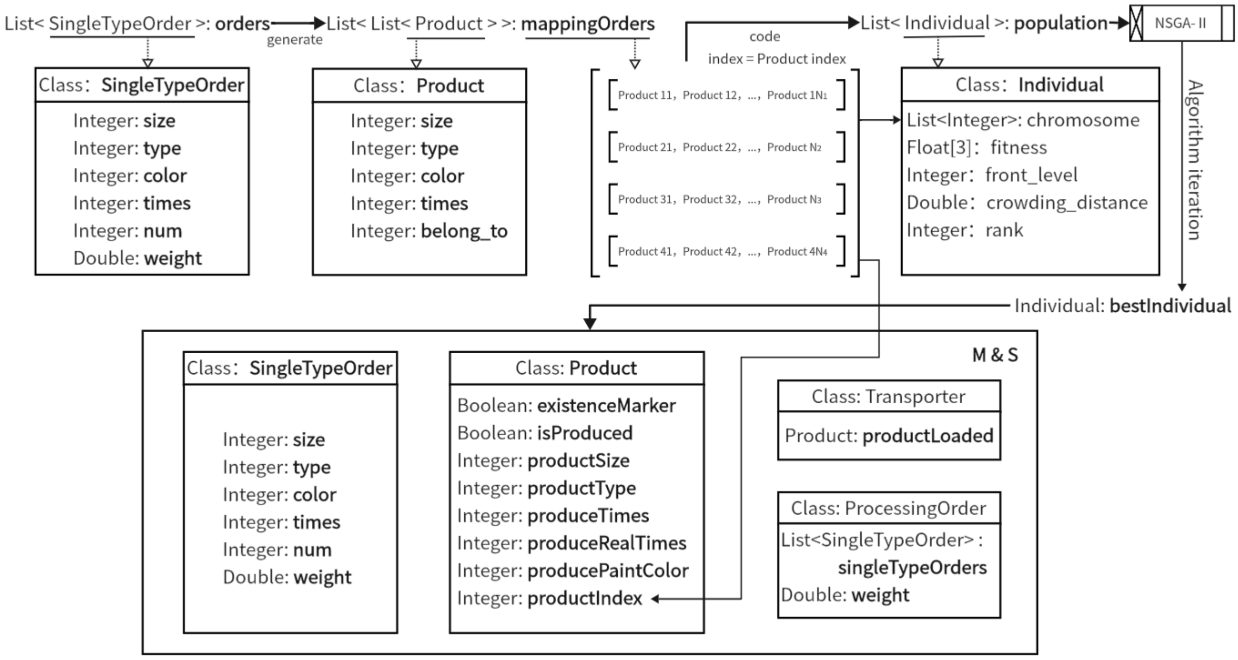

Figure 3. Data transformation diagram: data formats designed during database entry and propagated along fixed paths to support M&S.

Following this, the optimal solution is obtained using NSGA-II. To simulate real operational logic and construct a digital twin around the simulation, it is necessary to transmit the optimal solution to the simulation model. Real-time observation through the simulation program enhances the visibility of potential production issues.

3.2. Practical Application Context

This process involves loading different types of items onto hanging cars from multiple input points, taking them through various stages (including cleaning, masking, multiple spraying and leveling, and drying), and finally unloading by type, as illustrated in Figure 4. Detailed parameters will be provided in Sections 4.8 and 6.1.

Than we define the nature of the problem, determining the parameters and attributes required to build various models. The model construction is divided into three stages.

- First, the mathematical model is constructed, which will be used for the design and improvement of the algorithm.
- Next, the discrete system simulation model is constructed to verify the algorithm's results through simulation.
- Finally, the logical control model is constructed. These models, derived from the discrete system simulation model, will guide the operation of the suspension–chain–transportation system.

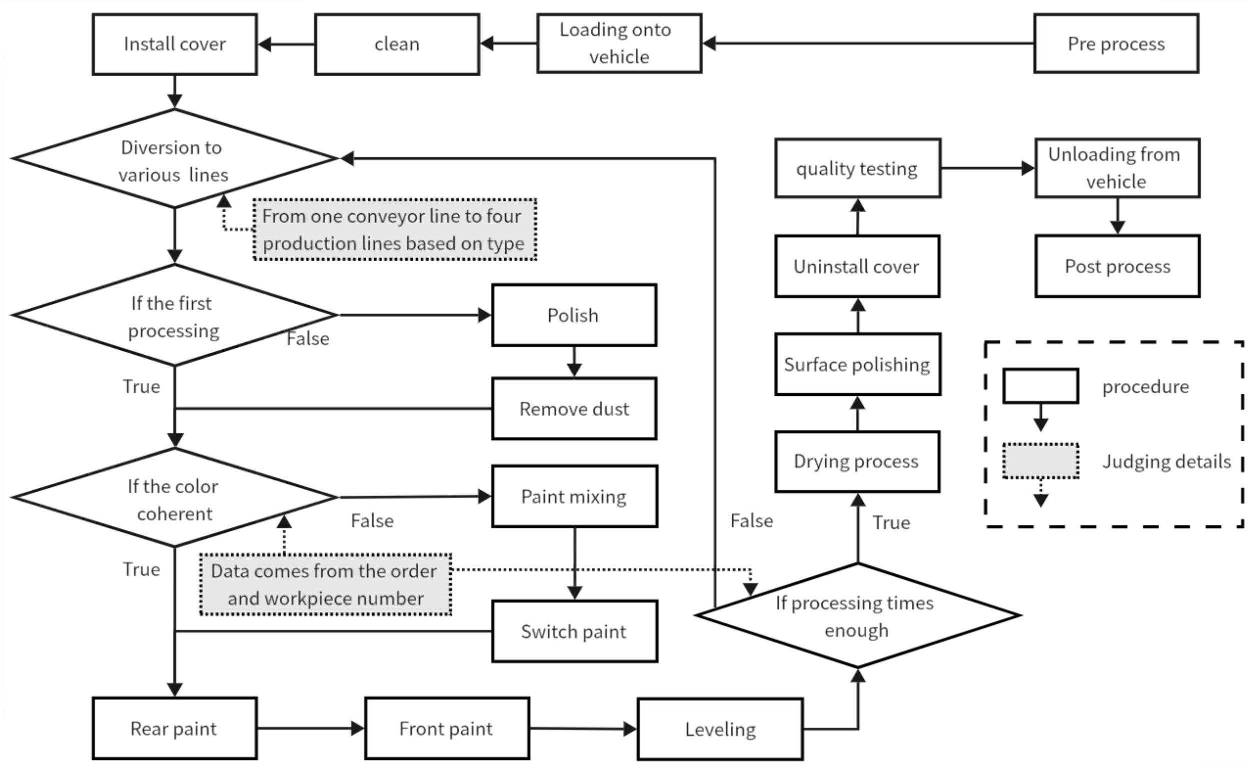

Figure 4. Flowchart of the production processes from loading to unloading of components, including masking, cleaning, diversion, leveling, surface treatment, drying, and unmasking.

Ultimately, these models will be integrated to form a complete order processor. The effectiveness of this system will be conveniently demonstrated by accompanying simulations.

Furthermore, as depicted in Figure 5, the data transmission of the CPSS is managed securely and efficiently. The physical layer includes carriers and rails. Information from both is transmitted through several edge devices, which also control the carriers and rails. The signals collected by these edge devices undergo data filtering, compression, and initial processing, followed by transmission to an Industrial Internet of Things gateway via Ethernet or Narrowband IoT (NB-IoT).

Subsequently, at the data layer, the database server is defined as a combination of persistent databases and task execution. It primarily handles bottom-layer data reception and top-layer feedback while interacting with edge devices, cloud servers, and order processors.

Lastly, at the virtual layer, the order data layer is constructed and integrated for transmission to the database. Based on database information, the corresponding ERP, MRP, CRP, etc., are formulated. The current production plan determines the set of orders to be processed, which serves as a parameter for fitness calculation and decoding in algorithms. The final results are returned to the database server, which issues instructions to complete the corresponding scheduling activities. Ultimately, this information is fed back to the cloud server to support the digital twin model.

This section primarily discussed the nature of the current problem and relevant technical approaches. Subsequent sections will provide more detailed designs for model implementation.

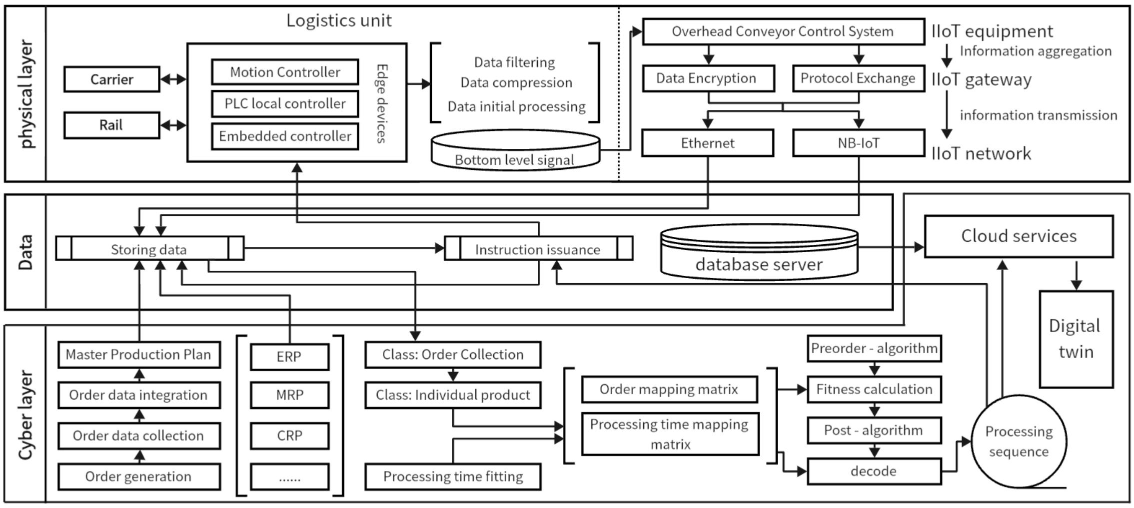

Figure 5. CPSS technical framework: enhancing the CPSS architecture from the perspective of adopted technologies and existing facilities.

4. Algorithm Model Establishment

This chapter primarily discusses the selection of operators within algorithms and their operational outcomes, alongside the algorithm refinement process.

4.1. Symbol Definitions

First, it is essential to clarify the parameters, hyperparameters, variables, and events within the mathematical model. This section focuses on defining each element individually, along with explaining their roles in the simulation model. These definitions will serve as the foundational control interface for the simulation model and subsequent algorithm optimizations.

Notably, the definitions provided here are specific to the parameters, hyperparameters, variables, and events required for the simulation model. Additional definitions will be introduced separately to the interface with algorithmic optimization, allowing for better decoupling and independent debugging of the model. The definitions of events, variables, and parameters are presented in Tables 1–3. Here, i represents a positive integer. Note that while other parameters may support the simulation model during actual runs, they are not included in the mathematical model establishment discussed herein.

Table 1. Event definition.

Name	Definition Range	Meaning
$checkHoldOnProduct$	$i \in [1, 8]$	If workpiece be loaded
$checkHoldOnOHC$	$i \in [1, 8]$	Allow Carriers Loading
$holdCheckOnInput$	-	If the total input sufficient
$testOfAutoControl$	-	Activate Algorithm Logic

Table 2. Variable definition.

Name	Definition Range	Data Type	Meaning
$ContentInput$	$i \in [1, 8]$	int	Current Input Lane with Carriages
$ContentInput1To8$	-	int	All Input Lanes with Carriages
$ContentInput1To4$	-	int	First, Four Inputs with Carriages
$ContentInput5To8$	-	int	Last Four Inputs with Carriages
$MoldChangeTimes_i$	$i \in [1, 8]$	int	Number of Changeovers per Station
$OrderOTime_i$	$i \in [1, OrdereNum]$	double	Output Time for Each Order
$OrderWeight_i$	$i \in [1, OrdereNum]$	double	Importance Index for Each Order

Table 3. Parameter definition.

Name	Definition Range	Data Type	Meaning
$CapacityI_i$	$i \in [1, 27]$	int	Input Section Capacity
$CapacityO_i$	$i \in [1, 6]$	int	Output Section Capacity
$CapacityC_i$	$i \in [1, 3]$	int	Cleaning and Masking Section Capacity
$CapacityP_i$	$i \in [1, 12]$	int	Processing Section Capacity
$CapacityR_i$	$i \in [1, 2]$	int	Multiple Processing Section Capacity
$CapacityD_i$	$i \in [1, 13]$	int	Drying Section Capacity
$ConveyorGap$	-	double	Safety Distance
$ArrivalInterval$	-	List<double>	Interval of Workpiece Arrivals
$TransportRate$	-	double	Conveyor Speed
$TransporterNum$	-	int	Number of Carriers
$ClearTime$	—	$f(type, size)$	Cleaning Processing Time
$CoverTime$	—	$f(type, size)$	Time of (un)Loading Masking
$PaintTime$	—	$f(type, size)$	Time of Spray Coating
$LoadTime_i$	$i \in [1, 8]$	double	Workpiece Loading Time
$UnloadTime_i$	$i \in [1, 8]$	double	Workpiece Unloading Time
$MoldChangeTime$	-	double	Changeover Time
$OrderNum$	-	int	Number of Orders
$ProductNum$	-	int	Number of Workpieces
$ContentI_i$	$i \in [1, 27]$	int	Input Section Current Inventory
$ContentO_i$	$i \in [1, 6]$	int	Output Section Current Inventory
$ContentC_i$	$i \in [1, 3]$	int	Cleaning and Masking Section Current Inventory
$ContentP_i$	$i \in [1, 12]$	int	Processing Section Current Inventory
$ContentR_i$	$i \in [1, 2]$	int	Multiple Processing Section Current Inventory
$ContentD_i$	$i \in [1, 13]$	int	Drying Section Current Inventory

4.2. Algorithm Framework Selection

NSGA-II is a framework primarily used for solving multiobjective problems. This framework employs dominance relationships instead of fitness as the criterion, and utilizes crowding distance to avoid premature convergence. Constructing Pareto fronts enhances search efficiency, and crowding selection serves as a balancing strategy, enabling NSGA-II to converge efficiently and stably.

The concept of non-dominated sorting, incorporated in the NSGA, adapts traditional GA for solving multiobjective problems. The key idea is the dominance relationship. When all objective functions of one individual are superior to those of another, the latter is said to be dominated by the former. An individual not dominated by any other individual is considered non-dominated individual. Multiple non-dominated individuals that do not dominate each other can exist, which are defined as the Pareto front. The Pareto hierarchy of an individual can be employed to evaluate its performance across multiple objectives.

However, individuals with similar Pareto rankings are frequently overly close in the solution space, causing the trapping of the algorithm in local optima. NSGA-II addresses this issue with elitism and crowding distance sorting. By calculating the sum of distances between an individual and other individuals in the solution space, its crowding distance is determined. Limiting the crowding distance of individuals effectively prevents the algorithm from being trapped in the local optima. The elitism strategy ensures that superior solutions are retained and not prematurely eliminated.

Here, NSGA-II is chosen for several reasons. First, it has already demonstrated mature applications and favorable results in production scheduling problems [52,53]. The current problem involves multiple objective functions and numerous factors due to production control, necessitating more objective functions to represent the desired outcomes of the study. As previously mentioned, NSGA-II is effective in solving multiobjective problems. Additionally, due to some constraints, a more flexible solution approach is required for the problem. Heuristic algorithms, such as NSGA-II, can be more adaptable in adjusting various operators to achieve different complex functions, which is precisely what this study requires.

Moreover, heuristic algorithms exhibit good compatibility with constraints, allowing developers to adjust operators to accommodate complex constraints in practical applications.

4.3. Encoding

The encoding in this study differ slightly from traditional scheduling problems. To enhance solution search efficiency, each individual is endowed with multiple chromosomes. Each chromosome's gene values represent workpiece IDs, the index of the gene values denotes the processing sequence, and chromosomes denote different processing lines.

4.4. Constraints

This study addresses two specific constraints. It can be formulated as follows, where m represents one-fourth of the capacity of the repeat processing waiting queue, and assuming $S = \{gen_1, gen_2, gen_3, \ldots, gen_n,\}$ represents the processing sequence of a certain line:

- Within the same production line, no two operations on any workpiece should occur consecutively more than once.
- Across all production lines, there should be no excessive distance between any two consecutive operations on any workpiece.

Those constraints can be expressed in the following form:

$$\forall i, j \quad (gen_i = gen_j \Rightarrow \{gen_k \mid i < k < j\} \cap \{gen_k \mid j < k < i\} = \emptyset) \tag{1}$$

$$\forall i, j \quad (a_i = a_j \Rightarrow |\{a_k \mid i < k < j\}| \leq m) \tag{2}$$

Subsequent sections will elaborate on the method through which these constraints are ensured in the solutions proposed. In addition, there are other general constraints to ensure the proper functioning of the model, as below:

$$Content_i < Capacity_i \tag{3}$$

Here, $Content_i$ includes $ContentI$, $ContentO$, $ContentC$, $ContentP$, $ContentR$, and $ContentD$. Similarly, $Capacity_i$ follows the same structure.

4.5. Objective Function

This study considers two objective functions.

1. Maximize the product of the weight w_i and the last output time t_{fi} of each order i, summed over all orders n:

$$Z_1 = \max\left\{\sum_{i=1}^{n}(w_i \times t_{fi})\right\} \tag{4}$$

   where w_i represents the weight of order i, and t_{fi} denotes the output time of the last workpiece of order i.
2. Minimize the number of tool changeovers across all production lines m, where t_c is the total time spent on tool changes per production line:

$$Z_2 = \min\left\{\sum_{j=1}^{m} t_c\right\} \tag{5}$$

   Here, t_c represents the total tool change time for production line j.

The choice of these objective functions is deliberate: while minimizing tool changeovers can be considered a lower-level semantic feature relative to the first objective, which inherently includes the goal of minimizing tool changeovers, frequent tool changes not only impact the processing times, but also involve material wastage. Thus, while the first objective function serves as a higher-level semantic feature for the second, it does not fully encompass it.

For the first objective function, K. Tamssaouet et al. proposed an optimization algorithm that combines evaluation criteria with different weights, demonstrating the feasibility of this approach [54]. The second objective function is more commonly encountered, frequently appearing in multi-objective optimization problems and flexible job shop scheduling problems. Yi Cheng, in his research, highlighted a current issue where mold changeover is often overlooked in most multi-objective scheduling problems [55]. To address this, they designed an objective function that incorporates mold changeover time, providing valuable guidance for multi-objective scheduling problems that involve complex factors.

Although NSGA-II primarily utilizes dominance relations and crowding distance for individual evaluations, for a clear comparison of individual performance and potentially as a criterion for future evaluations, arbitrary weights are temporarily assigned here. These weights can be adjusted according to specific enterprise requirements; the weights given here are 0.8 and 0.2.

Based on the models discussed above, the mathematical model for this study can be expressed as follows:

$$\begin{cases} \max\left\{Z_1 = \sum_{i=1}^{n}(w_i \times t_{fi})\right\}, \quad \min\left\{Z_2 = \sum_{j=1}^{m} t_c\right\} \\ s.t. \\ \forall i,j \quad (gen_i = gen_j \Rightarrow \{gen_k \mid i < k < j\} \cap \{gen_k \mid j < k < i\} = \varnothing) \\ \forall i,j \quad (a_i = a_j \Rightarrow |\{a_k \mid i < k < j\}| \leq m) \\ Content_i < Capacity_i \end{cases}$$

4.6. Operator Design

Considering that the algorithm must satisfy the two constraints mentioned earlier, certain operators must be improved accordingly. Various studies have addressed constraint adaptation in optimization planning. Some employ listeners or penalty functions to phase out infeasible solutions naturally. However, due to the vast solution space and sparse feasible solutions in our study, these methods are unsuitable. Another approach involves uniform distribution reference points, as in NSGA-III; however, even under constraints, searching remains challenging because of the expansive solution space. Therefore, certain operators are restructured into constraint-aware operators here, enabling automatic adaptation to constraints. This enhancement maximizes the heuristic algorithm's potential under constraints.

This section will focus on the refinement of each operator to ensure that the algorithm produces only feasible solutions. Figure 6 illustrates the primary processes of the algorithm.

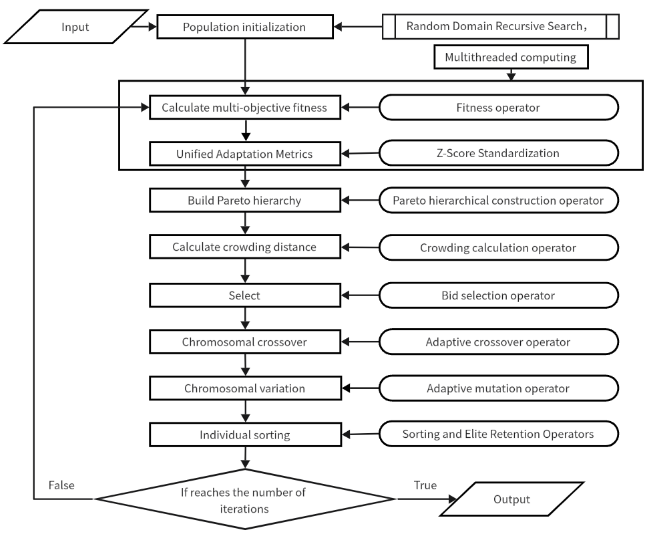

Figure 6. Algorithm flowchart: operators adjusted to ensure the generation of individuals adapting to current constraints.

4.6.1. Population Initialization

The population initialization directly influences the subsequent generation of feasible solutions. Typically, to generate feasible solutions that meet constraints, random generation is employed, followed by filtering or feasible solution transformations. However, random generation followed by filtering often has a low probability of producing feasible solutions, whereas feasible solution transformations struggle to ensure that the initial solutions are sufficiently randomized. Therefore, this study employs recursive backtracking specifically for generating feasible solutions. The pseudocode is depicted in Algorithm 1.

Algorithm 1 InsertGens

```
Require: sequence, gens, s_times_max, s_times_in_max
Ensure: t_sequence, t_gens, flag
child_nodes ← InsertGen(sequence, gen)
t_sequence ← sequence , gen ← gens[0] , t_gens ← gens[1 :]
while length of child_nodes == 0 do
  if s_times ≤ s_times_max or s_times_in < s_times_in_max then
    select_node ←, random.choice(child_nodes)
    t_sequence.insert(select_node, gen)
    if length(gens) == 1 then
      flag ← 1 , s_times ← s_times + 1
      return t_sequence, t_gens, flag
    else
      flag ← 2
      t_sequence, t_gens, flag ← InsertGens(t_sequence, t_gens, flag)
      if flag == 1 or −1 then
        return t_sequence, t_gens, flag
      else
        child_nodes.remove(select_node)
      end if
    end if
    s_times_in ← s_times_in + 1
  else
    return t_sequence, t_gens, −1
  end if
end while
return t_sequence, ghttps : //boardmix.cn/app/homeens, 3
```

When generating a random individual, all unique elements are first shuffled to form an initial sequence. The remaining elements that need to be duplicated are identified as "gens", and their insertion positions are evaluated sequentially by a random selection process.

However, situations where earlier insertions exhaust all possible insertion positions for subsequent elements may be encountered, necessitating a backtracking step.

Given the inherently exhaustive nature of this method, it can be time-intensive. To address this, a depth limit on the search is introduced. If the search exceeds this limit, the program is re-invoked to regenerate the initial solution. Empirical testing has shown that this effectively prevents excessively deep searches, significantly reducing the average time required to generate each solution.

4.6.2. Crossover

In crossover, we aim to preserve as many features of both parent chromosomes as possible. Traditional multi-point or single-point crossovers often result in invalid solutions, leading to the development of a constraint-aware crossover. As illustrated in Figure 7, this crossover method involves inheriting the order of the first occurrence of elements with the same repetition count from both parents. This technique ensures that a valid solution can be derived from another valid solution.

Assuming two parent chromosomes have been selected, we record the positions of each repeated element within each chromosome. Thereafter, we note the order of their first occurrences. Subsequently, the first occurrence orders of elements in the two parents are swapped. The repeated elements of one parent are replaced in the corresponding positions of the other parent according to this order.

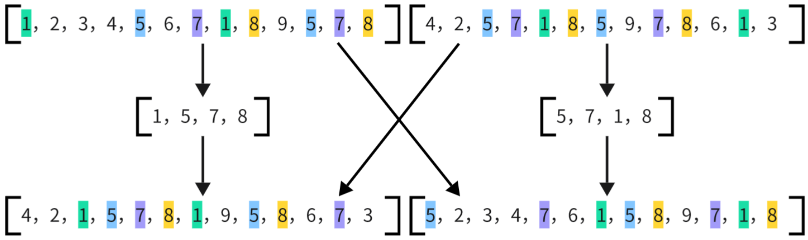

Figure 7. Constraint-aware crossover operator example: exchange the order of repeated elements based on their occurrence sequence with equal repetition counts.

4.6.3. Mutation

Mutation is another crucial step that requires operating on chromosomes. Considering the constraints, we adopt the mutation method illustrated in Figure 8. In each mutation, two elements with the same repetition count are selected, and their values at each position are swapped. This method, similar to the crossover operator, ensures that the resulting solutions remain valid after mutation.

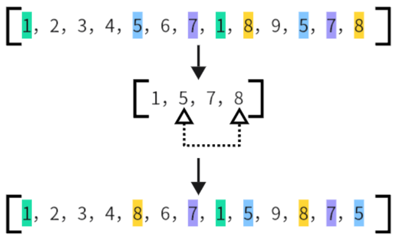

Figure 8. Constraint-aware mutation operator example: select any two elements with the same repetition count and swap their positions.

Furthermore, since the length of chromosomes varies in practical production, the number of mutations also varies with the length.

4.7. Fitting of Processing Time

In this study, a situation exists where there are over a thousand variations in product sizes and models. Therefore, individually measuring the processing time for each item is impractical.

Here, we assume that processing times follow a normal distribution. Through extensive statistical analysis and linear fitting, we obtained the mean and coefficient of variation values for the processing times based on product sizes and models. Using these parameters, we derived the processing times for each product model. Notably, due to the discrete nature of product types, fitting continuous functions is not meaningful. Products are categorized into eight classes, Types 1–8, and each is fitted separately. The results are depicted in the figures.

Parts of the fitted lines in Figures 9 and 10 are incomplete because the corresponding product sizes for certain categories do not exist. The results from linear regression for the mean processing times are shown in Figures 11 and 12. Notably, the processing time for painting is significantly greater than that for other processes.

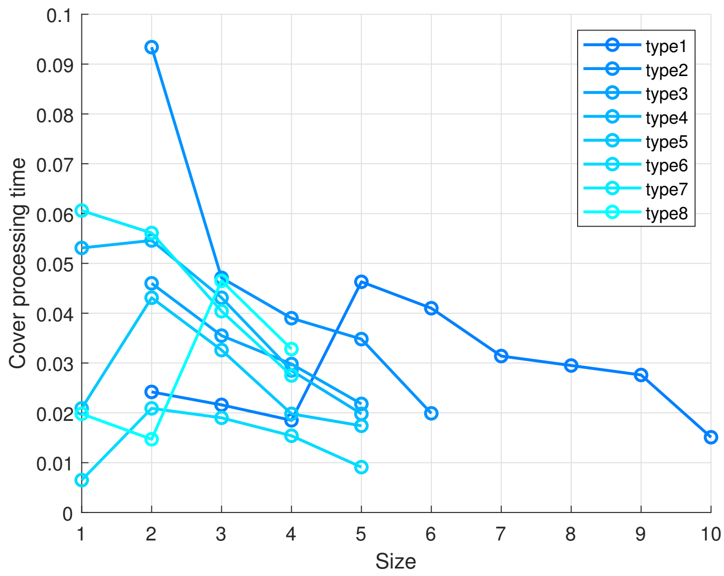

Figure 9. Variation coefficient fitting graph for cleaning and loading/unloading masking materials across different product types and sizes.

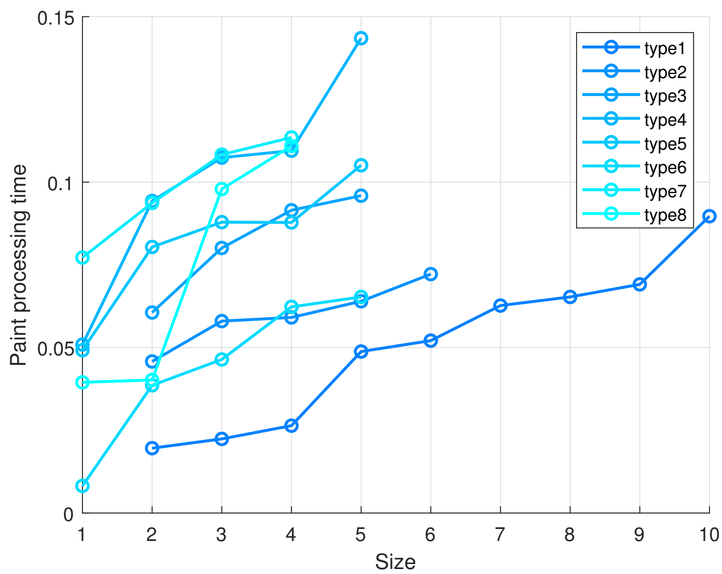

Figure 10. Variation coefficient fitting graph for spray coating process across different product types and sizes.

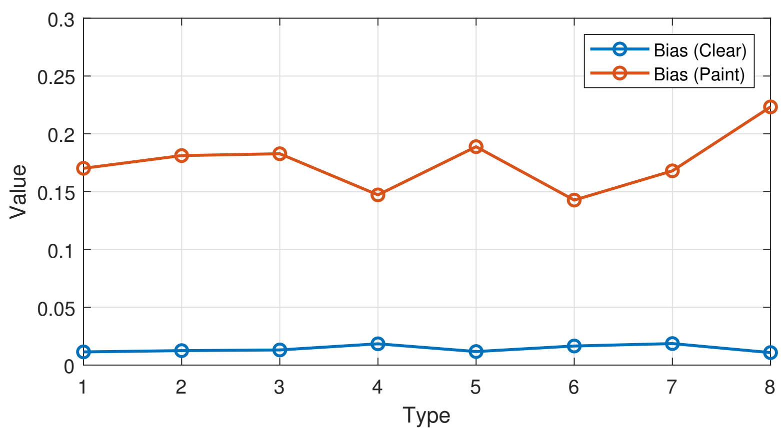

Figure 11. Linear fitting bias line graph for different product types.

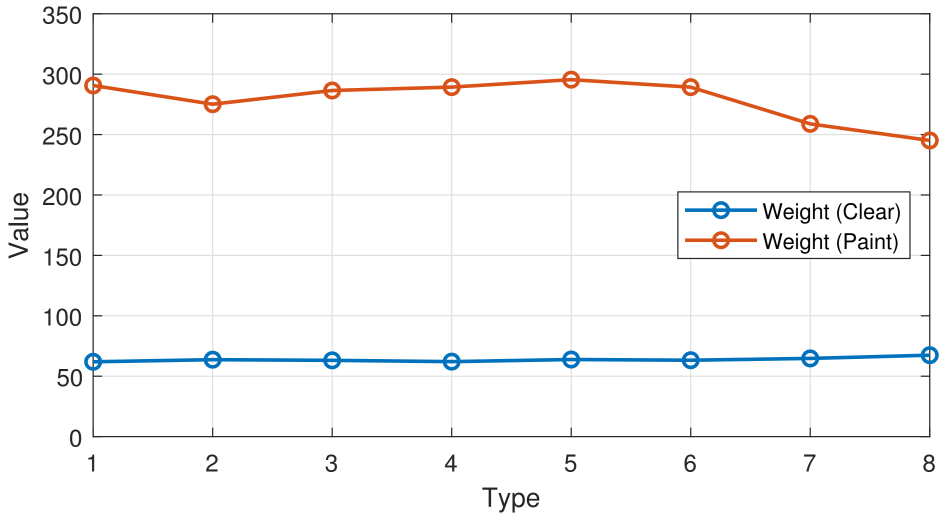

Figure 12. Linear fitting weight line graph for different product types.

4.8. Algorithm Convergence Results

To ensure the algorithm's convergence and reliability, multiple repeated experiments were conducted. Confidence analysis was performed on the results. Consistency across multiple runs indicates effective algorithm convergence. After conducting 10 consecutive experiments, the ratio of the confidence interval width to the mean was adopted as a measure of convergence accuracy. In the case study, 16 orders were inserted, totaling 125 different product types with various requirements; the detailed parameters are shown in Table 4. The final weighted objective function convergence for each experiment is presented in the table below.

Table 4. Assumed order parameters: several orders were generated using a random-number generator to obtain uniformly distributed parameters. Each order was assigned a weight uniformly at random by the generator.

Order	Color	Size	Type	Times	Num	Weight
1	3	8	1	1	6	0.60
2	1	10	1	2	3	0.65
3	1	5	2	2	9	2.90
4	2	6	2	2	11	0.68
5	3	4	3	2	14	0.53
6	1	3	3	1	11	0.07
7	1	3	4	2	6	1.70
8	2	1	4	1	3	0.22
9	3	1	5	1	6	1.80
10	1	3	5	2	12	0.01
11	3	1	6	2	3	0.47
12	3	3	6	2	9	0.48
13	1	1	7	2	11	1.55
14	1	3	7	1	3	0.48
15	3	1	8	2	14	1.24
16	3	1	8	2	4	0.43

Confidence analysis is a generally used method in experimental research, and the majority of randomized repeated experiments require confidence analysis to ensure the reliability of the result [56–58]. Confidence analysis is necessary. It has been confirmed and widely applied in the literature that confidence analysis ensures the credibility of conclusions.

Based on the data above, after excluding the outlier from the seventh experiment, there are nine effective samples to be calculated. Therefore, the degrees of freedom $d_f = 8$. Additionally, the confidence level is set to 0.9, resulting in a significance level of $\alpha = 0.1$. The mean and variance of these samples can be easily calculated as follows: $\bar{x} = -0.4880$, $s = 0.00041$.

$$\text{confidence interval} = \bar{x} \pm t_{\alpha/2,df} \cdot \frac{s}{\sqrt{n}} \tag{6}$$

Therefore, based on the experimental data in Table 5, the confidence interval can be calculated as $[-0.5011, -0.4775]$, with a width-to-mean ratio of 0.0482. From the results of the confidence analysis, the algorithm can be considered to have effectively converged. This conclusion is confirmed by examining the convergence curves from a randomly selected run, as depicted in Figure 13. Figure 14 shows a Gantt chart obtained by decoding the scheduling results from a selected experiment.

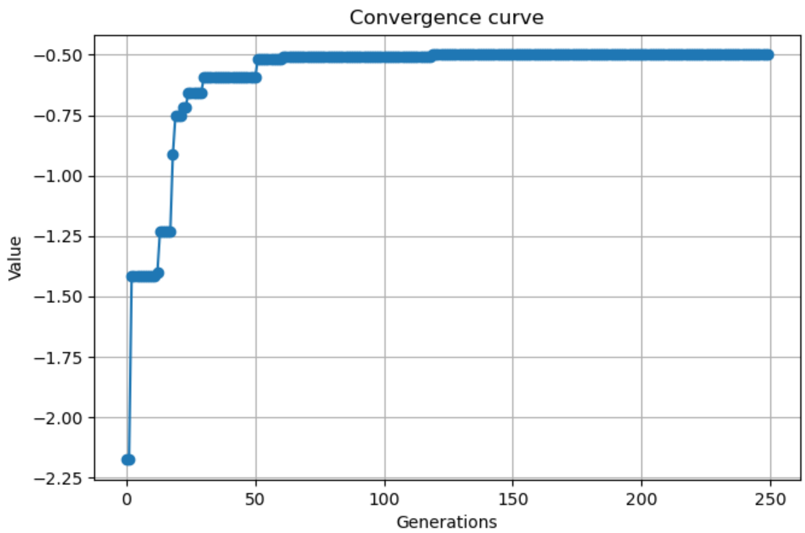

Figure 13. Convergence curve of a particular experiment: the algorithm tends to converge after 125 iterations.

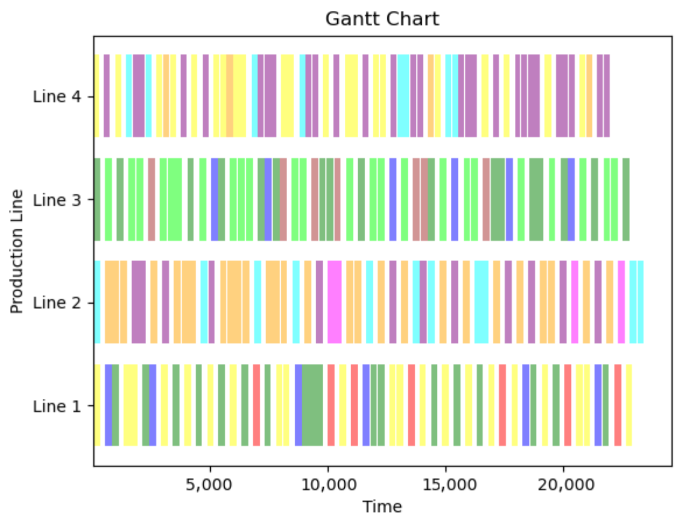

Figure 14. Results of a particular experiment. We decoded the results of a specific experiment and plotted them as a Gantt chart.

Table 5. Experimental results: twenty simulation experiments were conducted, and the weighted fitness of the optimal individual was recorded for each experiment.

Number of Experiments	Weighted Fitness
1	−0.5000
2	−0.4999
3	−0.5000
4	−0.4687
5	−0.4999
6	−0.5000
7	−0.5000
8	−0.2490
9	−0.4892
10	−0.4460

5. Simulation Model Establishment

To assess the practical performance of the algorithm, further simulation and modeling are essential. This chapter primarily outlines the key processes involved in constructing the model.

5.1. Simulation Fidelity

VV&A is a guiding principle employed to ensure that simulations achieve the expected results in modeling works. In this study, VV&A will be integrated throughout the entire research cycle.

5.1.1. Randomly Repeated Experiments

Randomized repeated experiments are critical for ensuring the reliability of the simulation model. They will be designed and implemented here. Additionally, a confidence interval analysis is conducted on the final results to ensure data credibility.

Regarding the simulation experiments, we will begin with 10 initial randomized simulations. Following a confidence interval analysis of the results, we will determine the number of subsequent experiments required.

5.1.2. Module Decoupling

Decoupling modules allows for easy modular debugging of the model. This approach facilitates identifying current issues and functionalities to be implemented throughout the model's development cycle.

Eight modules have been designed here, including a suspension car input module, workpiece input module, workpiece loading, cleaning and masking installation module, processing module, drying module, unmasking module, and workpiece unloading module.

Due to global requirements and software characteristics, an agent-based programming approach in Anylogic was adopted. Two main classes were designed: SuspensionCar and Workpiece. The attributes and behaviors relevant to these two agents were constructed. For simplicity, a detailed discussion is not provided here.

5.1.3. Constraints and Assumptions

There is no universally effective model. Clearly defining constraints and assumptions can better frame the model's boundaries and simplify it effectively. The purpose of these assumptions is to reduce workload without compromising operational outcomes.

- The suspension chain is replaced with a roller conveyor line, and turntables are employed at corners instead. Since their performance and parameters are identical and Anylogic does not have a pre-packaged suspension chain model, substituting with an existing roller conveyor line ensures equivalent effectiveness while considerably simplifying the workload.

- The tray's speed during operation is constant, and the acceleration of the tray on the conveyor belt is neglected. In production, trays move slowly; therefore, strict speed requirements are not necessary. Additionally, due to the significant acceleration of workpieces during movement, neglecting tray acceleration increases model stability.

To enhance the precision control of the model and prepare for potential future improvements, multiple listeners were designed within the simulation model. These listeners are utilized for scheduling conveyors to various loading points and controlling the merging and diverging of workpieces. While the listener mechanism may consume significant memory, it enables finer control granularity and robustness optimizations during each listening cycle, ensuring smoother model operation.

In summary, the entity model depicted in Figure 15 has been constructed. The corresponding logical model is presented in Figure 16.

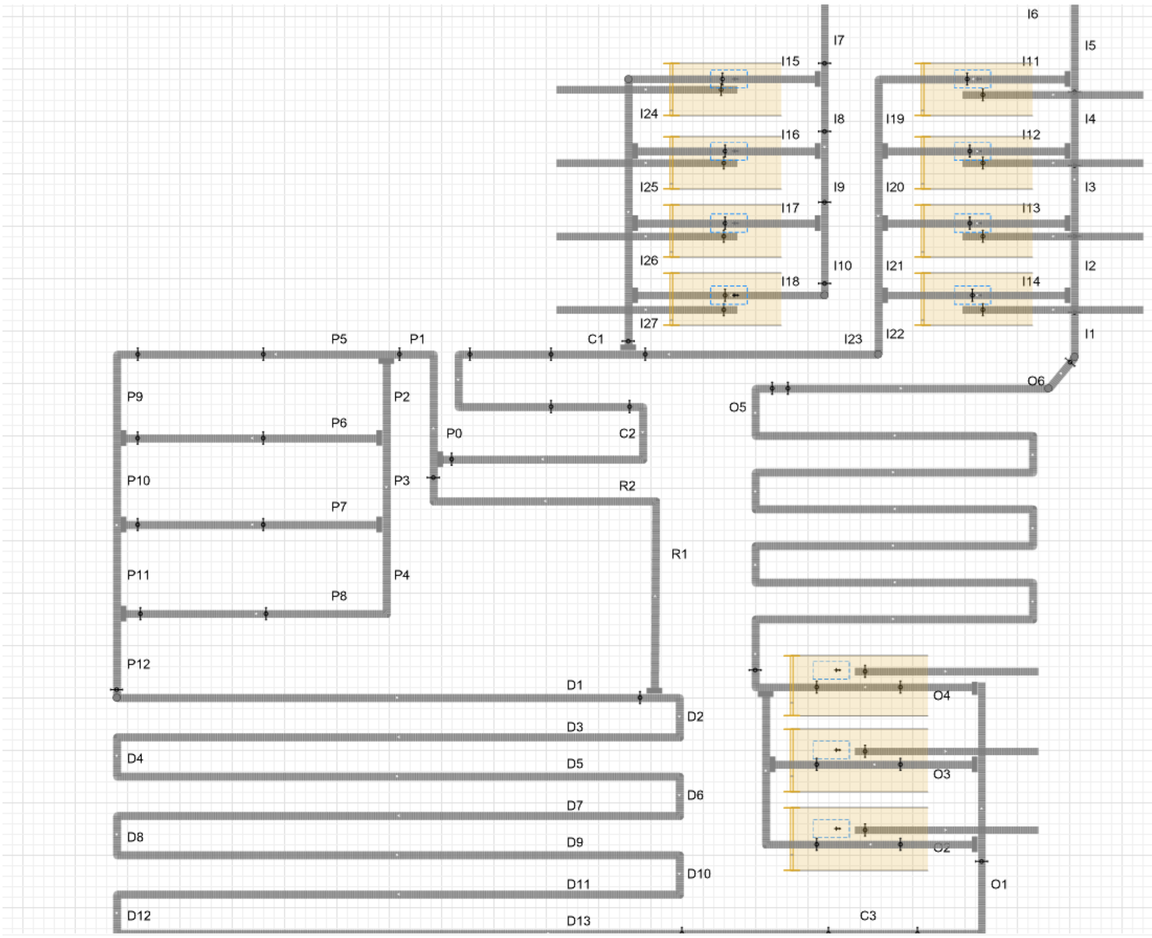

Figure 15. Entity modeling diagram: The image includes a partial representation of the entity model throughout the factory. Each conveyor segment is assigned a unique identifier.

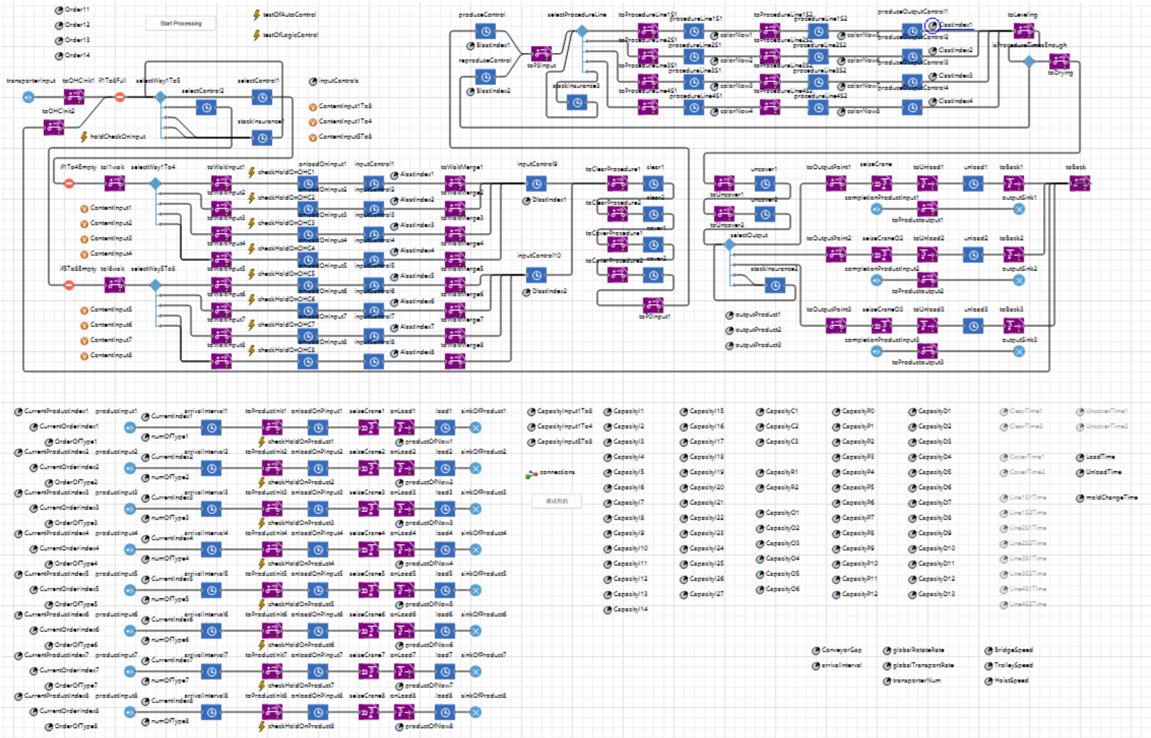

Figure 16. Overall logic diagram: this diagram encompasses the entire simulation flow, including definitions of logic modules, variables, events, and their interactions.

5.2. The Logic Control Scheme

It can be observed that the following five aspects require control:

- Output sequence from loading ports 1–4 to I22.
- Output sequence from loading ports 5–8 to I27.
- Output sequence from two loading-port convergence nodes into mainline C1.
- Output sequence from four processing lines to P12.
- Output sequence from the initial processing and repeat processing workpieces into main processing line P1.

By controlling the on/off states or input sequences of the above points, it indirectly controls the entire processing sequence. The control criterion is uniform throughout, which involves verifying whether incoming workpieces should be processed at the current stage.

6. Joint Simulation

This section primarily discusses the operational results of the integrated simulation model.

6.1. Simulation Parameters

The simulation parameters are based on research conducted in an actual factory setting.

For crane movement speed, actual speeds are disregarded, and a simple delay is adopted. Load and unload times are set to $LoadTime$ = $UnloadTime$ = 60 s. Mold change time is set to $moldChangeTime$ = 180 s.

Suspension chain operating speed is $TransportRate$ = 1 m/s. There are 50 transporters for the suspension chain, with a safety distance of $ConveyorGap$ = 0.8 m between them.

6.2. Experimental Design

Considering hardware limitations, simulating over 180 workpieces simultaneously is impractical. Therefore, a time-window simulation approach is adopted, where we focus the data-collection process on a specific time interval. Through optimized memory usage, the model runs smoothly up to the input of the 103rd workpiece. Therefore, the simulation experiment will terminate after inputting the 100th primary processed workpiece, representing a typical simulation run. The virtual simulation time is approximately 65,000 s.

Subsequently, under these conditions, 10 pre-experiments will be conducted to evaluate the outcomes. Based on these results and confidence interval analysis, the required number of subsequent experiments will be determined.

6.3. Simulation Result

As depicted in Figure 17, noticeable congestion relief is evident after the algorithm integration compared to before. While spraying remains the current bottleneck operation, deadlocks no longer occur. Given the stochastic processing times in the model, interference among waiting queues for parallel production lines has significantly eased compared to previous scenarios. The workstation-utilization rates across all production lines, balanced against varying production demands and conditions, exhibit a confidence interval of [0.7500, 0.7648] at a confidence level of $\alpha = 0.95$. This indicates considerable improvement from the initial conditions. Moreover, the average workstation-utilization rate increased from 0.3 to 0.7226. Since the calculation here is identical to the one in Section 4.8, it will not be reiterated.

The results demonstrate that this method effectively addresses the challenges faced by factories. The order set used in this study, consisting of 16 orders, reflects the typical number of orders companies process within a specific time frame. When handling larger order volumes, the algorithm offers even greater benefits, as manual decision-making incurs higher costs. Conversely, with lower order demand, management often consolidates production lines. The algorithm's parameters can be adjusted accordingly to accommodate these changes. Therefore, this model is well-suited to adapt to similar production methods across different companies.

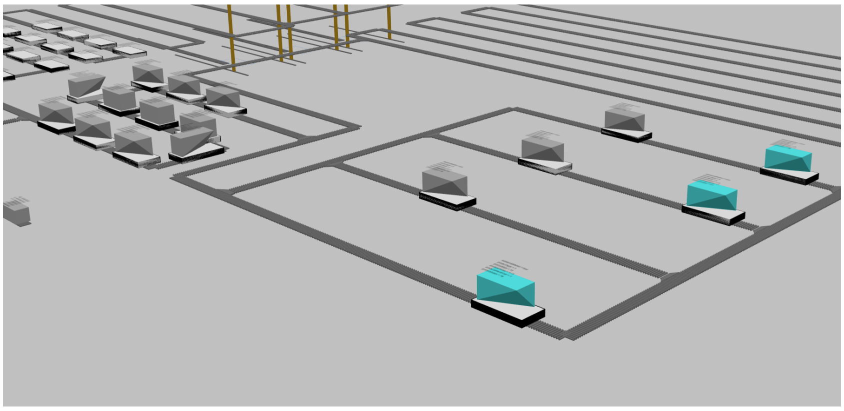

Figure 17. Three-dimensional simulation perspective after algorithm integration: Workpieces shown in the image are processed simultaneously.

7. Conclusions

This study leverages existing enterprise hardware and software infrastructure to design a CPSS. Detailed analysis and design of the data-processing aspects were conducted, and the problem was effectively solved using NSGA-II. Finally, an Anylogic simulation model was developed that integrates the algorithm with the simulation model. By controlling the simulation model with algorithmic results and corresponding logic, the integration of simulation and algorithm as a cyber layer in CPSS was demonstrated to be feasible.

Author Contributions: Conceptualization, J.L.; Methodology, W.Z. (Wenbin Zhao); Software, J.H.; Validation, W.Z. (Wenzhu Zhang); Formal analysis, J.H.; Investigation, W.Z. (Wenzhu Zhang); Data curation, J.H.; Writing—original draft, W.Z. (Wenbin Zhao); Writing—review & editing, W.Z. (Wenbin Zhao); Visualization, J.H.; Supervision, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. B24110200013.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Dafflon, B.; Moalla, N.; Ouzrout, Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: A literature review. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 2395–2412. [CrossRef]
2. Zamfirescu, C.B.; Pirvu, B.C.; Schlick, J.; Zuehlke, D. Preliminary Insights for an Anthropocentric Cyber-Physical Reference Architecture of the Smart Factory. *Stud. Inform. Control* **2013**, *22*, 269–278. [CrossRef]
3. Hermann, M.; Pentek, T.; Otto, B. Design Principles for Industry 4.0 Scenarios. 2015. Available online: http://www.snom.mb.tu-dortmund.de/cms/de/forschung/Arbeitsberichte/Design-Principles-for-Industrie-4_0-Scenarios.pdf (accessed on 11 May 2016).
4. Dai, K.; Zhu, Z.; Tang, Y.; Shen, G.; Li, X.; Sa, Y. Position synchronization tracking of multi-axis drive system using hierarchical sliding mode control. *J. Braz. Soc. Mech. Sci. Eng.* **2021**, *43*, 1–16. [CrossRef]
5. Strobel, J.; Sumpf, J.; Bartsch, R.; Golder, M. Real-time measuring and monitoring of relevant parameters in Complex chain conveyor systems. *innoTRAC J.* **2020**, *1*, 64–73. [CrossRef]
6. Kulkarni, A.J.; Kulkarni, T.M.; Mahadik, O.J.; Mahindrakar, P.V. Design of continuous loading vertical chain conveyor. *Int. J. Adv. Res. Ideas Innov. Technol.* **2018**, *4*, 465–469.
7. Issa, S.; Tu, Y. A survey in the resource-constrained project and multi-project scheduling problems. *J. Proj. Manag.* **2020**, *5*, 117–138. [CrossRef]
8. Guo, K.; Li, J.; Niu, M. Multi-agent interests service composition optimization in cloud manufacturing environment. *IEEE Access* **2023**, *11*, 53760–53771. [CrossRef]
9. Wang, Y.; Liu, P. Bi-level optimization of shared manufacturing service composition based on improved NSGA-II. *PLoS ONE* **2024**, *19*, e0303968. [CrossRef]
10. Jing, W.; Zhang, Y.; Chen, Y.; Zhang, H.; Huang, W. Cloud-edge collaboration composition and scheduling for flexible manufacturing service with a multi-population co-evolutionary algorithm. *Robot.-Comput.-Integr. Manuf.* **2024**, *90*, 102814. [CrossRef]
11. Sha, L.; Gopalakrishnan, S.; Liu, X.; Wang, Q. Cyber-physical systems: A new frontier. In Proceedings of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008), Taichung, Taiwan, 11–13 June 2008; pp. 1–9.
12. Wang, F.Y. The Emergence of Intelligent Enterprises: From CPS to CPSS. *IEEE Intell. Syst.* **2010**, *25*, 85–88. [CrossRef]
13. Lee, J.; Kundu, P. Integrated cyber-physical systems and industrial metaverse for remote manufacturing. *Manuf. Lett.* **2022**, *34*, 12–15. [CrossRef]
14. Monostori, L.; Kádár, B.; Bauernhansl, T.; Kondoh, S.; Kumara, S.; Reinhart, G.; Sauer, O.; Schuh, G.; Sihn, W.; Ueda, K. Cyber-physical systems in manufacturing. *CIRP Ann.* **2016**, *65*, 621–641. [CrossRef]
15. Kao, H.A.; Jin, W.; Siegel, D.; Lee, J. A Cyber Physical Interface for Automation Systems—Methodology and Examples. *Machines* **2015**, *3*, 93–106. [CrossRef]
16. Cinar, E.; Kalay, S.; Saricicek, I. A Predictive Maintenance System Design and Implementation for Intelligent Manufacturing. *Machines* **2022**, *10*, 1006. [CrossRef]
17. Fu, T.; Liu, S.; Li, P. Digital twin-driven smelting process management method for converter steelmaking. *J. Intell. Manuf.* **2024**, 1–17. [CrossRef]
18. Hoffmann, M.W.; Malakuti, S.; Grüner, S.; Finster, S.; Gebhardt, J.; Tan, R.; Schindler, T.; Gamer, T. Developing industrial cps: A multi-disciplinary challenge. *Sensors* **2021**, *21*, 1991. [CrossRef]
19. Yaacoub, J.P.A.; Salman, O.; Noura, H.N.; Kaaniche, N.; Chehab, A.; Malli, M. Cyber-physical systems security: Limitations, issues and future trends. *Microprocess. Microsyst.* **2020**, *77*, 103201. [CrossRef] [PubMed]
20. Tantawy, A.; Abdelwahed, S.; Erradi, A.; Shaban, K. Model-based risk assessment for cyber physical systems security. *Comput. Secur.* **2020**, *96*, 101864. [CrossRef]
21. Bernardi, S.; Gentile, U.; Marrone, S.; Merseguer, J.; Nardone, R. Security modelling and formal verification of survivability properties: Application to cyber–physical systems. *J. Syst. Softw.* **2021**, *171*, 110746. [CrossRef]
22. Yu, Z.; Gao, H.; Cong, X.; Wu, N.; Song, H.H. A Survey on Cyber–Physical Systems Security. *IEEE Internet Things J.* **2023**, *10*, 21670–21686. [CrossRef]
23. Lei, D. Multi-objective production scheduling: A survey. *Int. J. Adv. Manuf. Technol.* **2009**, *43*, 926–938. [CrossRef]
24. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
25. Zhang, X.; Li, C.; Zhang, J.; Yang, M.; Wu, W. Multi-objective evolutionary algorithm-enabled multi-stage collaborative scheduling for automotive production. *Comput. Ind. Eng.* **2024**, *191*, 110151. [CrossRef]

26. Yahui, W.; Ling, S.; Cai, Z.; Liuqiang, F.; Xiangjie, J. NSGA-II algorithm and application for multi-objective flexible workshop scheduling. *J. Algorithms Comput. Technol.* **2020**, *14*, 1748302620942467. [CrossRef]
27. Tikhamarine, Y.; Souag-Gamane, D.; Ahmed, A.N.; Kisi, O.; El-Shafie, A. Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey Wolf optimization (GWO) algorithm. *J. Hydrol.* **2020**, *582*, 124435. [CrossRef]
28. Zhang, Y.; Zhu, H.; Tang, D. An improved hybrid particle swarm optimization for multi-objective flexible job-shop scheduling problem. *Kybernetes* **2020**, *49*, 2873–2892. [CrossRef]
29. Ding, H.; Gu, X. Improved particle swarm optimization algorithm based novel encoding and decoding schemes for flexible job shop scheduling problem. *Comput. Oper. Res.* **2020**, *121*, 104951. [CrossRef]
30. Bo, L.; Chen, G.; Tao, N. An improved bacterial foraging optimization for multi-objective flexible job-shop scheduling problem. *J. Eur. Syst. Autom.* **2018**, *51*, 323.
31. Liang, X.; Du, Z. Genetic algorithm with simulated annealing for resolving job shop scheduling problem. In Proceedings of the 2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 20–22 November 2020; pp. 64–68.
32. Han, N.; Li, F.; Chen, S.; Zhang, K.; Feng, J. Research on Optimal Scheduling of Home Energy Management System Based on NSGA III Multi-Objective Optimization Algorithm. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2020; Volume 585, p. 012008.
33. Ma, Y.; Yu, L.; Zhang, G.; Lu, Z.; Wu, J. Source-load uncertainty-based multi-objective multi-energy complementary optimal scheduling. *Renew. Energy* **2023**, *219*, 119483. [CrossRef]
34. Fu, T.; Li, P.; Liu, S. An imbalanced small sample slab defect recognition method based on image generation. *J. Manuf. Process.* **2024**, *118*, 376–388. [CrossRef]
35. Bao, L.; Zheng, M.; Zhou, Q.; Gao, P.; Xu, Y.; Jiang, H. Multi-objective optimization of partition temperature of steel sheet by NSGA-II using response surface methodology. *Case Stud. Therm. Eng.* **2022**, *31*, 101818. [CrossRef]
36. Zhou, W.; Chen, F.; Ji, X.; Li, H.; Zhou, J. A Pareto-based discrete particle swarm optimization for parallel casting workshop scheduling problem with fuzzy processing time. *Knowl.-Based Syst.* **2022**, *256*, 109872. [CrossRef]
37. Li, S.; Zhang, M.; Wang, N.; Cao, R.; Zhang, Z.; Ji, Y.; Li, H.; Wang, H. Intelligent scheduling method for multi-machine cooperative operation based on NSGA-III and improved ant colony algorithm. *Comput. Electron. Agric.* **2023**, *204*, 107532. [CrossRef]
38. Fonseca, C.M.; Fleming, P.J. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Humans* **1998**, *28*, 26–37. [CrossRef]
39. Mezura-Montes, E.; Coello, C.A.C. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm Evol. Comput.* **2011**, *1*, 173–194. [CrossRef]
40. Ming, M.; Trivedi, A.; Wang, R.; Srinivasan, D.; Zhang, T. A dual-population-based evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans. Evol. Comput.* **2021**, *25*, 739–753. [CrossRef]
41. Qin, H.; Fan, P.; Tang, H.; Huang, P.; Fang, B.; Pan, S. An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint. *Comput. Ind. Eng.* **2019**, *128*, 458–476. [CrossRef]
42. Novara, F.M.; Novas, J.M.; Henning, G.P. A novel constraint programming model for large-scale scheduling problems in multiproduct multistage batch plants: Limited resources and campaign-based operation. *Comput. Chem. Eng.* **2016**, *93*, 101–117. [CrossRef]
43. Kui, C.; Li, B. Research on FJSP of improved particle swarm optimization algorithm considering transportation time. *J. Syst. Simul.* **2021**, *33*, 845–853.
44. Peng, C.; Li, Z.; Zhong, H.; Li, X.; Lin, A.; Liao, Y. Research on Multi-Objective Scheduling Algorithm of Job Shop Considering Limited Storage and Transportation Capacity. *IEEE Access* **2023**, *11*, 94252–94280. [CrossRef]
45. Dan, Y.; Liu, G.; Fu, Y. Optimized flowshop scheduling for precast production considering process connection and blocking. *Autom. Constr.* **2021**, *125*, 103575. [CrossRef]
46. Akbar, M.; Irohara, T. NSGA-II variants for solving a social-conscious dual resource-constrained scheduling problem. *Expert Syst. Appl.* **2020**, *162*, 113754. [CrossRef]
47. Garcia Marquez Basaldua, J.C. Cyber-Physical Production Systems and Their Practical Integration and Application with Simio Software. Master's Thesis, University of Texas at El Paso, El Paso, TX, USA, 2023.
48. Balci, O. Verification, validation, and accreditation. In Proceedings of the 1998 Winter Simulation Conference. Proceedings (cat. no. 98ch36274), Washington, DC, USA, 13–16 December 1998; Volume 1, pp. 41–48.
49. *IEEE 1730.2™-2022*; IEEE Recommended Practice for Verification, Validation and Acceptance/Accreditation of a Distributed Simulation: An Overlay to the Distributed Simulation Engineering and Execution Process. The Institute of Electrical and Electronics Engineers, Inc.: New York, NY, USA, 2022.
50. *SISO-GUIDE-001.2-2013*; Guide for Generic Methodology for Verification and Validation (GMVV) to Support Acceptance of Models, Simulations, and Data, GM-VV Volume 2: Implementation Guide. Simulation Interoperability Standards Organization, Inc.: Orlando, FL, USA, 2013.
51. Wang, Y.; Li, J.; Hongbo, S.; Li, Y.; Akhtar, F.; Imran, A. A survey on VV&A of large-scale simulations. *Int. J. Crowd Sci.* **2019**, *3*, 63–86.
52. Yuan, M.; Li, Y.; Zhang, L.; Pei, F. Research on intelligent workshop resource scheduling method based on improved NSGA-II algorithm. *Robot.-Comput.-Integr. Manuf.* **2021**, *71*, 102141. [CrossRef]

53. Shi, S.; Xiong, H. Solving the multi-objective job shop scheduling problems with overtime consideration by an enhanced NSGA-II. *Comput. Ind. Eng.* **2024**, *190*, 110001. [CrossRef]
54. Tamssaouet, K.; Dauzère-Pérès, S.; Knopp, S.; Bitar, A.; Yugma, C. Multiobjective optimization for complex flexible job-shop scheduling problems. *Eur. J. Oper. Res.* **2022**, *296*, 87–100. [CrossRef]
55. Cheng, Y.; Xie, Z.; Xin, Y.; Chen, K.; Zarei, R. Flexible job shop scheduling method for optimizing mold resource setup time. *IEEE Access* **2024**, *12*, 33486–33503. [CrossRef]
56. Calzarossa, M.C.; Della Vedova, M.L.; Massari, L.; Nebbione, G.; Tessera, D. Multi-objective optimization of deadline and budget-aware workflow scheduling in uncertain clouds. *IEEE Access* **2021**, *9*, 89891–89905. [CrossRef]
57. Amiri, M.F.; Behnamian, J. Multi-objective green flowshop scheduling problem under uncertainty: Estimation of distribution algorithm. *J. Clean. Prod.* **2020**, *251*, 119734. [CrossRef]
58. Yao, S.; Jiang, Z.; Li, N.; Zhang, H.; Geng, N. A multi-objective dynamic scheduling approach using multiple attribute decision making in semiconductor manufacturing. *Int. J. Prod. Econ.* **2011**, *130*, 125–133. [CrossRef]