buildings

MDPI

Article

Balancing Efficiency and Ergonomics-Aware Pallet Layout Optimization in Precast Concrete Production

Junyong Liang [1], Yongzhong Zhang [1], Shicen Liu [1], Bing Yi [1], Baigang Du [2] and Hansong Jiang [3,*]

[1] School of Civil Engineering, Sichuan University of Science and Engineering, Zigong 643000, China
[2] School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China
[3] Financial Investment Evaluation Center of Zigong High-Tech Zone, Zigong 643000, China
* Correspondence: jianghansong2026@outlook.com

Abstract

Pallet layout in precast concrete component production influences resource consumption, workstation takt balance, and operational safety. Yet, existing studies mostly focus on improving space utilization with limited consideration of the ergonomic efficiency losses and safety risks associated with workers climbing onto pallets. To address this issue, this study formulates a tri-objective mixed-integer programming model that jointly minimizes the total number of activated pallets, the variance of pallet-loaded areas, and the frequency of worker climbing operations. To resolve multi-objective conflicts across high-dimensional discrete search spaces, a non-dominated sorting genetic algorithm guided by Q-Learning, termed QL-NSGA-II, is developed to dynamically schedule the crossover and objective-oriented mutation strategies online to obtain Pareto-optimal solutions. Computational experiments across nine test configurations derived from an actual production order of 127 precast concrete components demonstrate that in the large-scale representative case of 127 components, the proposed algorithm improves the hypervolume metric by 8.1% to 14.2%, reduces the inverted generational distance by 25.3% to 41.3%, improves the spacing metric by 34.7%, and increases the number of non-dominated solutions by 28.37% to 49.16% compared with benchmark algorithms. Sensitivity analysis further indicates that expanding the effective working range from 450 mm to 800 mm can inherently reduce the frequency of worker climbing operations from 298 to 220, representing a 26.17% decrease, thereby supporting a balanced trade-off among production economy, line balancing, and occupational health and safety.

Keywords: precast concrete components; pallet layout; ergonomics; Q-learning; QL-NSGA-II

Check for updates

Academic Editor: Rafiq Muhammad Choudhry

Received: 22 October 2026
Revised: 20 November 2026
Accepted: 24 November 2026
Published: 26 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the continuous advancement of construction industrialization, prefabricated buildings have become an important direction for the transformation and upgrading of the modern construction industry, owing to their advantages in energy saving, emissions reduction, quality control, and shorter construction periods. In the prefabricated building supply chain, factory production of precast concrete (PC) components is a crucial link [1,2]. Modern PC component plants generally adopt a flow-line mode for centralized production [3]. In this context, pallet layout, as the first process in component production, directly determines the resource consumption and operational efficiency of subsequent processes such as steel reinforcement binding, concrete casting, and steam curing [4].

In real production scenarios, a high-quality pallet layout scheme often needs to balance production demands across multiple dimensions [5]. The first is the economic dimension.

The more compactly components are arranged on a pallet, the fewer pallets are required, which can effectively reduce equipment turnover and curing energy costs [6]. The second is the production-line-balance dimension. If the areas of components carried by different pallets differ considerably, the processing times of different pallets at the same workstation may fluctuate notably, leading to takt imbalance and workstation congestion on the flow line [4]. The third is the human factors ergonomics dimension, which is often overlooked in traditional research. Since a component production line usually allows workers to operate only along the two long sides of a pallet, when components, especially those with embedded wire boxes, are placed in the central blind zone of the pallet, workers frequently have to climb onto the pallet to perform cross-over operations. This increases operational safety hazards and notably reduces the production efficiency of a single workstation. Therefore, pallet layout is essentially a typical multi-objective combinatorial optimization problem.

At the algorithmic level, the two-dimensional layout problem with complex geometric avoidance and orthogonal rotation constraints is NP-hard [7]. As the number of components increases, conventional exact solvers often struggle to converge within a reasonable time. Existing metaheuristic algorithms, such as genetic algorithms and simulated annealing, usually adopt fixed-weight scalarization when dealing with multi-objective layout. This approach is easily influenced by the subjective preferences of decision makers, and in high-dimensional solution spaces, traditional blind neighborhood perturbation tends to trap the search in local extrema [3]. In recent years, although data-driven methods such as reinforcement learning have begun to show promise in combinatorial optimization, how to integrate them more deeply with traditional heuristic search to adaptively address multi-objective conflicts in layout remains to be explored further.

In summary, existing research seldom considers human ergonomics in precast component pallet layout, and traditional algorithms are prone to the curse of dimensionality when facing large-scale search spaces. The main research contributions of this paper can be summarized as follows.

(1) A mixed-integer programming model for precast component pallet layout that integrates human ergonomic constraints is constructed. By introducing rotation transformation equations for embedded part coordinates and an accessibility determination mechanism for operations on both sides of the pallet, the physical accessibility of the worker operating space is elevated from qualitative experience to quantifiable mathematical constraints, thereby addressing the long-standing absence of the human factors ergonomics dimension in existing layout models.

(2) This study overcomes the limitation of traditional layout research, which treats space utilization as a single orientation. The tri-objective optimization system jointly incorporates the number of pallets used (reflecting economic performance), the area load variance (reflecting production line balance), and the frequency of climbing onto the pallet for elevated operations (reflecting ergonomic performance). By using fast non-dominated sorting and crowding-distance elitist preservation instead of traditional fixed-weight scalarization evaluation, the method achieves an unbiased approximation and uniform diversity along the tri-objective Pareto conflict front.

(3) A non-dominated sorting genetic algorithm with Q-Learning adaptive guidance, termed QL-NSGA-II, is proposed. Q-Learning is embedded within the evolutionary generations of NSGA-II as an online generation-level parameter controller. By perceiving the real-time macroscopic population state across three dimensions—namely convergence stagnation, crowding-distance diversity, and objective bottleneck—it dynamically schedules six reproduction strategies (combining two crossover operators and five general or objective-oriented mutation operators) guided by a two-dimensional

stepwise reward function based on Pareto front updates and offspring survival rates, ensuring a smooth transition from global exploration to local exploitation.

The remaining sections of this paper are organized as follows. Section 2 systematically reviews the literature on prefabricated component production, scheduling algorithms, and two-dimensional layout optimization, and clarifies this paper's research entry point. Section 3 describes the pallet layout problem in detail and constructs a multi-objective mixed-integer programming model that integrates human factors ergonomic constraints. Section 4 elaborates on the overall architecture, encoding and decoding schemes, genetic operator design, and Q-Learning adaptive parameter control mechanism of the proposed QL-NSGA-II algorithm. Section 5 verifies the effectiveness of the proposed model and algorithm through computational experiments and case analyses. Finally, Section 6 summarizes the research work of the full paper and discusses future research directions.

2. Literature Review

2.1. PC Production

Regarding production planning, capacity simulation, and lean production, early studies constructed a decision-making framework for production planning in precast plants. By distinguishing long and short production series and exploring the configuration logic of multi-mold systems, these studies provided a theoretical reference for subsequent research [8]. Subsequently, some scholars developed a capacity planning model based on computer simulation, enabling managers to evaluate the specific impact of different management strategies on capacity before production [9]. Further research integrated neural networks and genetic algorithms to support the production planning of customized components, which helped shorten delivery lead times and accumulate production experience [10]. In the process of introducing lean manufacturing concepts, related studies systematically identified eight types of waste in production and proposed seven corresponding implementation principles [11]. A lean production system constructed based on the logic of constant work-in-process was verified through simulation to save labor costs [12]. Value stream mapping tools were also applied to identify non-value-added activities in exterior wall panel production lines, thereby achieving line rebalancing [13]. In addition, by integrating component complexity assessment, discrete event simulation, and lean analysis to improve production planning, the quantitative level of lean transformation decisions was enhanced [14].

With respect to information management, quality inspection, and production pattern mining, a precast production management system based on technologies such as radio frequency identification was developed, achieving information collection and tracking throughout the full production cycle of components [15]. A method integrating data cleaning, data mining, and simulation optimization further enabled real-time planning control of panelized housing production lines [16]. An automated dimensional quality assurance method for full-scale components combining building information modeling and 3D laser scanning achieved millimeter-level inspection accuracy, improving work efficiency compared with manual inspection [17]. A learning curve model established based on a large amount of production data determined the training time thresholds for various key processes, providing a quantitative basis for worker skill management [18]. The construction of a complexity index system for precast components and its analysis results verified that complexity has a notable negative impact on production efficiency [19]. Meanwhile, research incorporating the cross-training costs of multi-skilled workers into production optimization models revealed the trade-off relationship between production flexibility and cost [20].

2.2. Production Scheduling Algorithms for PC

In terms of static scheduling modeling and optimization algorithms, related studies abstracted the production of heterogeneous components under specialized process organization as a flow shop sequencing model, adopted genetic algorithms to optimize makespan and tardiness penalties, and verified the advantages of intelligent algorithms over classical heuristic rules [21–23]. Subsequently, researchers applied genetic algorithms under resource constraints to improve precast plant output [24] and further extended them to handle mixed production scenarios [25]. A multi-objective flow shop model established for customized components comprehensively considered total flow time, equipment idle time, and early and late delivery penalties, and conducted sensitivity analysis on parameters such as mold quantity [26]. After constraint programming was introduced into the field of precast production scheduling, favorable computational performance was obtained by leveraging the separation of the model and the solver [27]. A multi-objective decision support system developed by incorporating inter-station buffers into the model showed that considering buffer factors helps obtain more feasible scheduling schemes [28]. For research on the integration of concrete mix proportion selection and scheduling under parallel molds, the proposed heuristic method can effectively shorten computation time while maintaining relatively good solution quality [29]. The scheduling model was subsequently extended to a multi-production-line environment and incorporated constraints for frequent switching of component types [30]. The established mixed-integer linear programming model showed positive effects in simultaneously minimizing storage and delay time, effectively alleviating delay phenomena [31]. For a model established for distributed multi-factory just-in-time scheduling, effective solutions were achieved by designing a hybrid tabu and iterative greedy algorithm [32].

With regard to dynamic rescheduling and intelligent algorithms, the rescheduling problem in multi-production-line environments has received attention, with research considering both adjustment costs and scheduling stability [33]. A proposed hybrid model of two-level simulation and genetic algorithm characterized the uncertainty of processing times [34]. A dynamic flow shop model considering early delivery, urgent order insertion, and order cancelation showed potential for saving production costs in practical cases [35]. A hybrid rescheduling optimization model proposed for equipment failures aimed to further enhance scheduling robustness [36]. The adoption of a proactive scheduling strategy can better cope with the uncertainty of contractor progress under certain conditions [37]. After reinforcement learning methods were applied to precast component production scheduling, they demonstrated performance superior to traditional methods in controlling tardiness indicators, and the proposed action-oriented deep reinforcement learning method, by defining the set of unscheduled components as the action space, possesses a certain cross-scale generalization capability [38]. Furthermore, recent advancements have demonstrated that integrating deep learning or reinforcement learning paradigms with classical multi-objective evolutionary algorithms can significantly enhance state representation and adaptive search capabilities [39].

2.3. PC Layout Optimization

In the area of pallet layout, cutting, and mold configuration, studies have shown that the multi-period production planning of precast beams is isomorphic to the classical multi-period cutting stock problem. By establishing and solving an integer linear programming model, a connection between precast production scheduling and two-dimensional layout theory was established [40]. Research using genetic algorithms to minimize the number of molds used explored bin packing optimization in the mold dimension while considering both makespan and tardiness penalties [41]. A mixed-integer linear programming

model, based on the concepts of precast configuration and component grouping, optimized the utilization of complex molds and effectively controlled production costs [42]. A component layout scheme optimized with pallet utilization as the objective further enriched the application scenarios of the typical two-dimensional rectangular layout problem [43]. Research combining backtracking and association rules to optimize pallet combination layout considered both combination frequency and space utilization [44]. A multi-objective genetic algorithm was used to automatically optimize steel mold dimensions and assembly schemes [45]. A collision-free reinforcement layout was generated using multi-agent collaboration and particle swarm optimization, extending layout ideas to the steel bar cutting and placement stage [11].

Concerning component stacking and yard layout, research considering on-site assembly sequence to optimize the stacking scheme of composite slabs effectively reduced secondary sorting operations through joint modeling of sequencing and stacking [46]. The developed related simulation model can optimize yard layout and dynamic product allocation, providing an early reference for yard space layout simulation [47]. Research incorporating hoisting efficiency and safety distance constraints into component yard layout optimization achieved coordination between space utilization and operational safety [48]. Overall, research in this direction builds on a two-dimensional rectangular layout and bin packing theory as its core and unfolds along a chain from pallet layout, mold configuration, and finished product stacking to yard layout. However, heuristic layout for irregular components and online dynamic layout remain relatively limited and can serve as directions for further exploration.

2.4. Research Gap

A comprehensive review of the preceding literature on PC production, workshop scheduling, and PC layout optimization shows that existing research has yielded substantial results across various subfields. However, when these results are applied in an integrated manner to real precast component flow-line production scenarios, three critical limitations remain.

First, real-world production orders typically involve multi-specification PC components with strict geometric clearance and orthogonal rotation constraints; inefficient two-dimensional spatial arrangements directly increase the number of activated pallets, thereby increasing mold occupancy, equipment turnover, and steam-curing energy costs. Second, the pallet layout scheme not only determines pallet turnover costs, but also affects the variance in the area load carried by each activated pallet, which directly relates to processing-time fluctuations and takt-time balance at subsequent serial workstations on the flow line. Finally, existing pallet layout models mostly treat maximizing pallet space utilization or minimizing the number of pallets as the sole dominant optimization objective. In actual prefabricated factories, workers are constrained by pallet dimensions and the physical layout of workstations and can usually perform production operations only in the narrow aisles along the two long sides of the pallet. The existing literature rarely associates the geometric coordinates of component-embedded parts, such as wire boxes, with the physical accessibility of the worker's operating space. As a result, generated PC layout schemes often place embedded parts in the pallet's central blind zone, forcing workers to frequently climb onto the pallet for crossover operations, which can create safety hazards and increase time consumption due to poor ergonomics.

3. Model of Pallet Layout in PC Production (PL-PC)

3.1. Model Assumptions

To ensure that the model remains practically applicable while being mathematically tractable, the following basic assumptions are proposed based on typical production conditions in a PC component factory.

Assumption 1: All PC components to be laid out are rectangular panels.

Assumption 2: Each PC component may be placed on a pallet in four orthogonal rotations, with rotation angles $\theta \in \{0°, 90°, 180°, 270°\}$.

Assumption 3: All pallets used on the PC production line have the same effective working surface dimensions $L \times W$.

Assumption 4: Workers can perform production operations only from the aisles along the two long sides of the pallet, that is, the upper and lower sides along the width direction W, and cannot enter the pallet area from the short sides. The effective working depth from either long side toward the interior of the pallet is δ. This parameter is determined according to ergonomic standards and field measurement data and is treated as a known constant in the model.

Assumption 5: The number of embedded parts on each type of component and their relative position coordinates in the component body coordinate system is known input information. The geometric dimensions of the embedded parts themselves are relatively small compared with the component as a whole, and their occupied area is ignored in the layout calculation.

3.2. Problem Description

The problem studied in this paper arises from the actual production operations at a prefabricated concrete component factory. In a modern PC component flow line, as shown in Figure 1, the pallet is the basic physical carrier that supports components through all production processes. After entering the flow line, each pallet passes sequentially through six serial workstations: side mold assembly, steel mesh placement and embedded part installation, concrete casting and vibration compaction, steam curing, demolding, and PC component repair. Pallet layout refers to determining, before components are formally put into production, the allocation scheme and geometric arrangement of a batch of components on each pallet. The numbers contained within the gray rectangles in the pallets represent the types of PC components. That is, which components are loaded together on the same pallet, and at which coordinate position each component is placed on the pallet surface. Obviously, the quality of the layout scheme directly affects the utilization efficiency of pallet resources and the takt time balance of each workstation on the flow line.

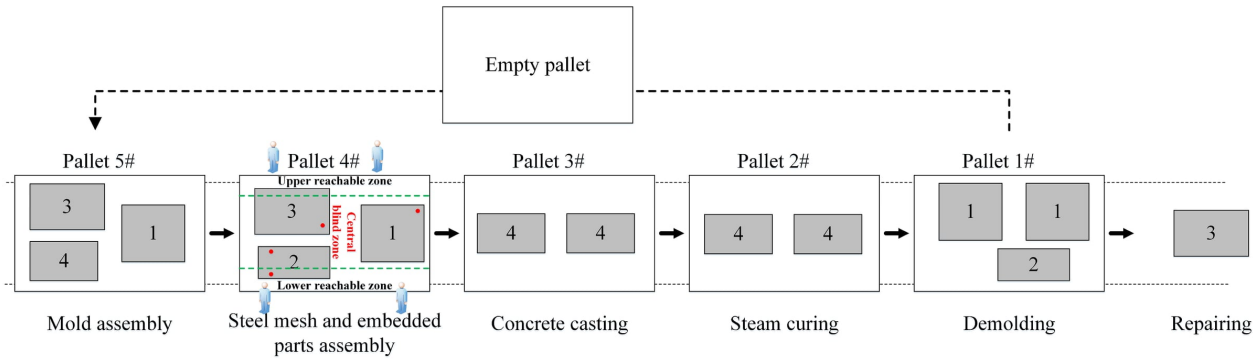


Figure 1. PC production processes on the assembly line.

Meanwhile, embedded parts, such as wire boxes, need to be manually fixed to the steel mesh by workers and cannot be handled by a movable robotic arm, as in the side mold assembly of PC components. This leads to a factor in pallet layout that is easily overlooked but has a notable impact on production efficiency, namely, the accessibility of the worker's operating space on the pallet.

In a typical PC component production line layout, the pallet moves along its length on a track. Workers can only stand in the aisles along the two long sides of the pallet to perform operations and cannot enter from the short sides. Constrained by human arm span and safe operating posture, there is a physical limit to the effective working depth from either long side of the pallet toward the interior. Consequently, two comfortable working zones are formed along the long sides of the pallet in the width direction: one in the bottom-side region and the other in the top-side region. The range between the two comfortable working zones constitutes a central blind zone that is difficult for workers to reach naturally from either side. When an embedded part on a certain component needs to be installed in the central blind zone on the pallet, workers cannot complete the placement and wiring of the wire box in a normal standing posture in the aisle and must climb onto the pallet surface to perform elevated operations.

As shown in Figure 2, climbing operations not only increase occupational safety risks, such as falls and strains, but also reduce PC production efficiency due to physical exertion and fatigue from repeated bending. Therefore, reducing worker climbing operations in PC production is significant for improving production efficiency and reducing occupational safety risks.

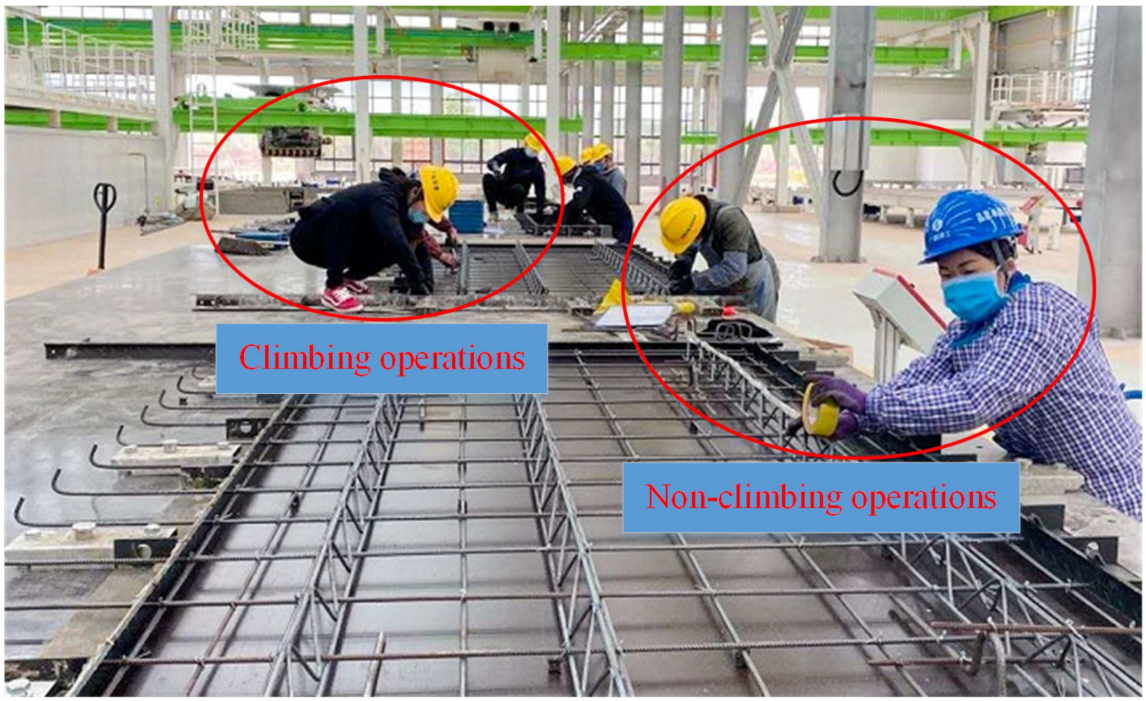


Figure 2. Climbing and non-climbing operations in PC production.

3.3. Mathematical Model for PL-PC

Based on the above assumptions, the pallet layout scheme is required to satisfy the following constraints. Mathematical model symbols are detailed in Appendix A.

$$\sum_{j \in J} z_{ip,j} = 1, \ \forall i \in N, p \in P_i \tag{1}$$

$$z_{ip,j} \le u_j,\ \forall i \in N, p \in P_i, j \in J \tag{2}$$

$$S_j = \sum_{i \in N} \sum_{p \in P_i} A_i \cdot z_{ip,j},\ \forall j \in J \tag{3}$$

$$\theta_{ip}^{0} + \theta_{ip}^{90} + \theta_{ip}^{180} + \theta_{ip}^{270} = 1,\ \forall i \in N, p \in P_i \tag{4}$$

$$\begin{aligned} \hat{l}_{ip} &= l_i \cdot \left(\theta_{ip}^{0} + \theta_{ip}^{180}\right) + w_i \cdot \left(\theta_{ip}^{90} + \theta_{ip}^{270}\right), \quad \forall i \in N, p \in P_i \\ \hat{w}_{ip} &= w_i \cdot \left(\theta_{ip}^{0} + \theta_{ip}^{180}\right) + l_i \cdot \left(\theta_{ip}^{90} + \theta_{ip}^{270}\right), \quad \forall i \in N, p \in P_i \end{aligned} \tag{5}$$

$$\begin{aligned} x_{ip} &\ge d/2,\ \forall i \in N, p \in P_i \\ y_{ip} &\ge d/2,\ \forall i \in N, p \in P_i \end{aligned} \tag{6}$$

$$\begin{aligned} x_{ip} + \hat{l}_{ip} + d/2 &\le L + M_x\left(1 - z_{ip,j}\right), \quad \forall i \in N, p \in P_i, j \in J \\ y_{ip} + \hat{w}_{ip} + d/2 &\le W + M_y\left(1 - z_{ip,j}\right), \quad \forall i \in N, p \in P_i, j \in J \end{aligned} \tag{7}$$

$$\begin{aligned} x_{ip} + \hat{l}_{ip} + d &\le x_{mr} + M_x\left(1 - \phi_{ip,mr}^{L}\right),\ \forall i, m \in N, p, r \in P_i \\ x_{mr} + \hat{l}_{mr} + d &\le x_{ip} + M_x\left(1 - \phi_{ip,mr}^{R}\right),\ \forall i, m \in N, p, r \in P_i \\ y_{ip} + \hat{w}_{ip} + d &\le y_{mr} + M_y\left(1 - \phi_{ip,mr}^{B}\right),\ \forall i, m \in N, p, r \in P_i \\ y_{mr} + \hat{w}_{mr} + d &\le y_{ip} + M_y\left(1 - \phi_{ip,mr}^{T}\right),\ \forall i, m \in N, p, r \in P_i \end{aligned} \tag{8}$$

$$\phi_{ip,mr}^{L} + \phi_{ip,mr}^{R} + \phi_{ip,mr}^{B} + \phi_{ip,mr}^{T} \ge z_{ip,j} + z_{mr,j} - 1,\ \forall j \in J \tag{9}$$

$$\begin{aligned} gx_{ip,k} &= x_{ip} + px_{ik} \cdot \theta_{ip}^{0} + (w_i - py_{ik}) \cdot \theta_{ip}^{90} + (l_i - px_{ik}) \cdot \theta_{ip}^{180} + py_{ik} \cdot \theta_{ip}^{270},\ \forall i \in N, p \in P_i, k \in K_i \\ gy_{ip,k} &= y_{ip} + py_{ik} \cdot \theta_{ip}^{0} + px_{ik} \cdot \theta_{ip}^{90} + (w_i - py_{ik}) \cdot \theta_{ip}^{180} + (l_i - px_{ik}) \cdot \theta_{ip}^{270},\ \forall i \in N, p \in P_i, k \in K_i \end{aligned} \tag{10}$$

$$\begin{aligned} gy_{ip,k} - \delta &\le M \cdot \xi_{ip,k}^{B},\ \forall i \in N, p \in P_i, k \in K_i \\ \delta - gy_{ip,k} &\le M \cdot (1 - \xi_{ip,k}^{B}),\ \forall i \in N, p \in P_i, k \in K_i \end{aligned} \tag{11}$$

$$\begin{aligned} (W - \delta) - gy_{ip,k} &\le M \cdot \xi_{ip,k}^{T},\ \forall i \in N, p \in P_i, k \in K_i \\ gy_{ip,k} - (W - \delta) &\le M \cdot \left(1 - \xi_{ip,k}^{T}\right),\ \forall i \in N, p \in P_i, k \in K_i \end{aligned} \tag{12}$$

$$C_{ip,k} \ge \xi_{ip,k}^{B} + \xi_{ip,k}^{T} - 1,\ \forall i \in N, p \in P_i, k \in K_i \tag{13}$$

$$\begin{aligned} C_{ip,k} &\le \xi_{ip,k}^{T},\ \forall i \in N, p \in P_i, k \in K_i \\ C_{ip,k} &\le \xi_{ip,k}^{B},\ \forall i \in N, p \in P_i, k \in K_i \end{aligned} \tag{14}$$

$$x_{ip}, y_{ip}, gx_{ip,k}, gy_{ip,k} \ge 0 \tag{15}$$

Equation (1) requires that each component in each production batch be assigned to only one pallet. Equation (2) specifies that when a component is placed on a pallet, the pallet is set to the activated state. Equation (3) is used to sum the areas of all components on the same pallet to calculate the actual occupied area of that pallet.

Equation (4) requires that each component selects one of the four orthogonal rotation postures only. Equation (5) updates the actual projected length and width of a component on the pallet according to the selected rotation angle: the dimensions remain unchanged at 0 degrees and 180 degrees, while the length and width are exchanged at 90 degrees and 270 degrees. Equation (6) requires that the left side and the bottom side of a component keep an operational clearance of at least $d/2$ from the pallet edge. Equation (7) requires that the right side and the top side of a component also keep an operational clearance of at least $d/2$ from the pallet edge. If the component is not assigned to the current pallet, a large M is used to deactivate this constraint. Equation (8) defines the relative position relationship between any two components on the same pallet and requires that their production clearance be at least d. Equation (9) specifies that if and only if two components are placed on the same pallet, at least one of the four relative position relationships holds, so as to avoid physical overlap between components.

Equation (10) indicates that after a component is rotated in four directions, the design coordinates of embedded parts are converted into absolute coordinates on the pallet through

linear mapping based on the rotation angle indicator variable. Equations (11) and (12) are used to determine the ergonomic operation type when workers operate on both sides of the precast concrete production line. Equation (11) forces the binary variable $\xi^B_{ip,k}$ to 1 if the y-coordinate of the embedded part $gy_{ip,k}$ exceeds the comfortable reach δ from the bottom edge. The Big-M constant ensures that when $gy_{ip,k} \leq \delta$, $\xi^B_{ip,k}$ can take the value of 0 to minimize the objective. Similarly, Equation (12) forces $\xi^T_{ip,k} = 1$ if the embedded part is out of reach from the top aisle ($gy_{ip,k} \leq W - \delta$). Equation (13) defines the triggering logic of the operation type: for the operation of the same embedded part, it is judged as needing to be completed on the pallet if and only if it exceeds the comfortable working area on both the upper and lower sides. Equation (14) specifies that if the embedded part can be operated from either side, it is recognized as being able to be completed on the lower side of the pallet. Equation (15) defines the value ranges of the decision variables.

Based on the above constraints, Equation (16) aims to minimize the total number of activated pallets to reduce production resource consumption, thereby directly reducing mold occupancy and tooling turnover costs. Equation (17) seeks to minimize the variance of the loaded area across all activated pallets, thereby leveling the production load and rhythm of each workstation. Equation (18) aims to minimize the total frequency of operations on the pallets, reducing labor intensity and safety risks.

$$\min f_1 = \sum_{j \in J} u_j \tag{16}$$

$$\min f_2 = \sqrt{\frac{1}{\sum_{j \in J} u_j} \sum_{j \in J, u_j = 1} \left(S_j - \frac{A_{total}}{\sum_{j \in J} u_j} \right)^2} \tag{17}$$

$$\min f_3 = \sum_{i \in N} \sum_{p \in P_i} \sum_{k \in K_i} C_{ip,k} \tag{18}$$

There is an inherent trade-off among the above three objectives. For example, excessively pursuing high pallet utilization (optimizing f_1) may force some components to be placed in the central area of the pallet, causing embedded parts to fall into the blind zone and increasing the number of climbing operations (worsening f_3). Conversely, concentrating components along the two side edges of the pallet to reduce climbing operations may lead to greater differentiation in the area load among pallets (worsening f_2). Therefore, the goal of this paper is not to find a single optimal solution, but to obtain a Pareto-optimal solution set with different gradients across the three objectives, providing factory decision-makers with schemes that can be flexibly selected according to actual production priorities.

4. QL-NSGA-II for PL-PC

4.1. Advantages of the QL-NSGA-II Algorithm

To solve the tri-objective pallet layout optimization model constructed in Section 3, this paper proposes an adaptive evolutionary algorithm that closely couples Q-Learning with NSGA-II, termed Q-Learning-guided NSGA-II and abbreviated as QL-NSGA-II. In standard NSGA-II, the crossover and mutation operator types and probabilities are usually set at algorithm startup and remain fixed throughout the evolutionary process. However, the requirements for genetic operators differ notably across evolutionary stages. In the early search stage, the population has not yet sufficiently covered all regions of the objective space, and a stronger perturbation is needed to accelerate global exploration. In the mid-search stage, some objectives may approach their ideal values earlier, while others still have considerable room for improvement, requiring targeted objective-oriented operators for focused local optimization. In the late search stage, the Pareto front tends to stabilize,

and high-intensity crossover recombination is needed to finely exploit the genetic complementarity among high-quality individuals. If the same operator configuration is adopted throughout, it is difficult to adapt to such dynamically changing search requirements over the evolutionary process.

To this end, this paper introduces Q-Learning as a generation-level parameter controller. Before each generation begins, the Q-Learning agent selects, based on the current population's convergence state, diversity level, and the objective bottleneck's characteristics, the most suitable set of parameter configurations from a pre-designed library of iterative strategies. This set covers the crossover and mutation operator types and their probabilities, and is used for that generation's NSGA-II iteration. After each generation, Q-Learning receives an immediate reward based on the actual improvement of the Pareto front and updates its policy value table. Through online learning accumulated over hundreds or thousands of generations, the agent can gradually acquire an empirical mapping of which reproduction strategy to adopt under which population states, thereby enabling the transition of operators and parameters from static presetting to dynamic adaptive adjustment.

4.2. Encoding and Decoding Schemes

4.2.1. Encoding Scheme

This paper adopts a two-segment chromosome encoding scheme consisting of a PC component sequence and a rotation vector. The chromosome of each individual is composed of two parts, as shown in Equation (19). The sequence segment is a permutation of a specified number of PC components that determines the order in which components are placed on the pallet. The rotation segment specifies the randomly determined rotation direction of each PC component.

For example, suppose the components to be laid out include 3 of type A, 2 of type B, and 2 of type C, for a total of $T = 7$ elements. A valid chromosome encoding can be expressed as shown in Equation (20). The sequence segment indicates that A_1 is placed first, then B_2, and so on. The rotation segment indicates that A_1 is placed at 0°, B_2 at 90°, and so forth.

$$Encoding\ Chromsome = [\underbrace{\pi(1), \pi(2), \ldots, \pi(T)}_{PC\ component\ sequence} \mid \underbrace{\theta_1, \theta_2, \ldots, \theta_T}_{Rotation}] \tag{19}$$

$$\left[A_1, B_2, A_3, A_2, C_1, C_2, B_1 \mid 0^\circ, 90^\circ, 180^\circ, 0^\circ, 270^\circ, 0^\circ, 90^\circ\right] \tag{20}$$

4.2.2. Decoding Scheme

Given a chromosome, the decoder sequentially places PC components onto pallets according to the encoding sequence, thereby completing the mapping from the chromosome to a complete pallet layout. The specific decoding algorithm is described as follows:

Step 1. Initialization: open the first pallet ($j = 1$), and all pallets are empty.

Step 2. In the encoding sequence order, take the current component $\pi(t)$, read its rotation angle θ_t, and calculate the envelope dimensions after rotation ($\hat{l}_t$, $\hat{w}_t$). Try to place the current PC component into each activated pallet in turn using the Bottom-Left placement heuristic [49], which searches for the lowest and leftmost feasible coordinate position (x_t, y_t) that simultaneously satisfies the pallet boundary clearance, inter-component safety clearance, and non-overlap constraints. If placement succeeds on some pallet j, record the pallet number j and the coordinate (x_t, y_t), and proceed to the next component. If no existing pallet can accommodate it after traversing all activated pallets, activate a new pallet $j = j + 1$ and place the component at the bottom-left starting position ($d/2$, $d/2$) of the new pallet.

Step 3. After all PC components have been placed on the pallets, output the complete pallet layout scheme.

4.3. NSGA-II Evolution Process

4.3.1. Initial Population Generation

A hybrid strategy that combines greedy guidance and random perturbation is used to generate an initial population. First, one greedy individual is created. Its sequence segment is arranged in descending order of PC component area, and its rotation angle is uniformly set to 0°. This strategy simulates the classical first-fit decreasing heuristic and tends to generate compact schemes with fewer pallets, providing the population with an initial reference point of relatively high space utilization.

Second, the remaining individuals are generated. For each random individual, the sequence segment of the encoding is a random permutation of all components, and the rotation angle is randomly selected from the four orthogonal orientations. This strategy introduces sufficient structural diversity into the population, enabling the initial population to cover different regions of the solution space.

4.3.2. Crossover Operator Design

In this paper, two crossover operators that act on the component sequence are designed, each with different preferences for preserving genetic information, for Q-Learning to select and invoke at different evolutionary stages. The crossover operation processes both the sequence segment and the rotation segment.

(1) Order Crossover (OX)

OX preserves the relative precedence relationships of components in the parent individuals and is an exploitation-oriented crossover. The specific steps are as follows:

Step 1. Let the two parent sequences be P_A and P_B. Randomly select two crossover points, pos1 and pos2, on the sequence, with pos1 < pos2.

Step 2. Copy the segment of P_A located within the interval [pos1, pos2] directly into the same positions of the offspring sequence P_O.

Step 3. Starting from the position after pos2 in P_B, with cyclic wrapping, read the components that have not yet appeared in the offspring in their order of appearance, and fill them into the empty positions after pos2 in the offspring sequence P_O, also with cyclic wrapping.

Step 4. Inheritance rule for rotation angles: components within the segment interval [pos1, pos2] inherit the rotation angles of the corresponding components in P_A, while components in the remaining positions inherit the rotation angles of the corresponding components in P_B.

Figure 3 shows the detailed process of performing OX on the two parents according to the above steps, with the two crossover points located at positions 3 and 5. The other offspring individual, obtained by directly copying the segment from PB, can be generated using the same procedure and is therefore not described again. The characteristic of OX is that it preserves the placement order information of a continuous segment of components from P_A, while supplementing the arrangement of the remaining components through the relative order in P_B. It is suitable for fine recombination of high-quality sequence segments in the middle and later stages of the search.

(2) Partially Mapped Crossover (PMX)

PMX generates offspring by establishing positional mapping relationships between parents, allowing greater mixing of the two parents' structural information. It is an exploration-oriented crossover. The specific steps are as follows:

Step 1. Randomly select two crossover points, pos1 and pos2, and copy the segment of P_A within the interval [pos1, pos2] to the same positions of the offspring P_O.

Step 2. Based on the correspondence between P_A and P_B within this interval, establish a set of mapping pairs between components.

Step 3. For positions in the offspring that have not yet been filled, read the component at the corresponding position from P_B. If this component has not yet appeared in the offspring, place it directly. If it already exists, that is, a conflict occurs, trace along the mapping chain to find a substitute component that has not yet appeared and place it.

Step 4. The inheritance rule for rotation angles is the same as that of OX: components within the segment interval inherit the rotation angles of P_A, and the rest inherit the rotation angles of P_B.

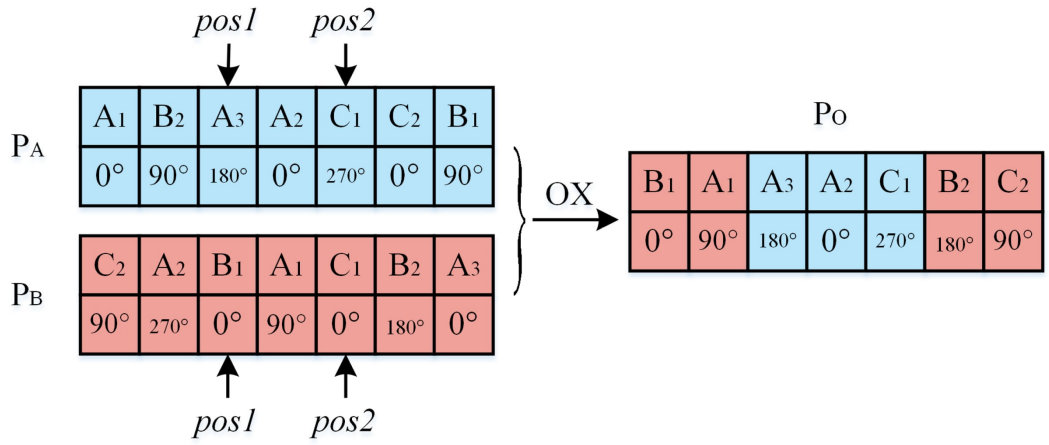


Figure 3. Order crossover operation.

Figure 4 illustrates the detailed process of performing PMX on two parents according to the above steps, with the two crossover points located at positions 3 and 5. PMX breaks the continuous segment structure of the parents and mixes the two parents' genes at finer granularity through mapping relationships. It can produce offspring individuals with relatively large structural differences, making it suitable for global exploration in the early search stage or when population diversity is insufficient.

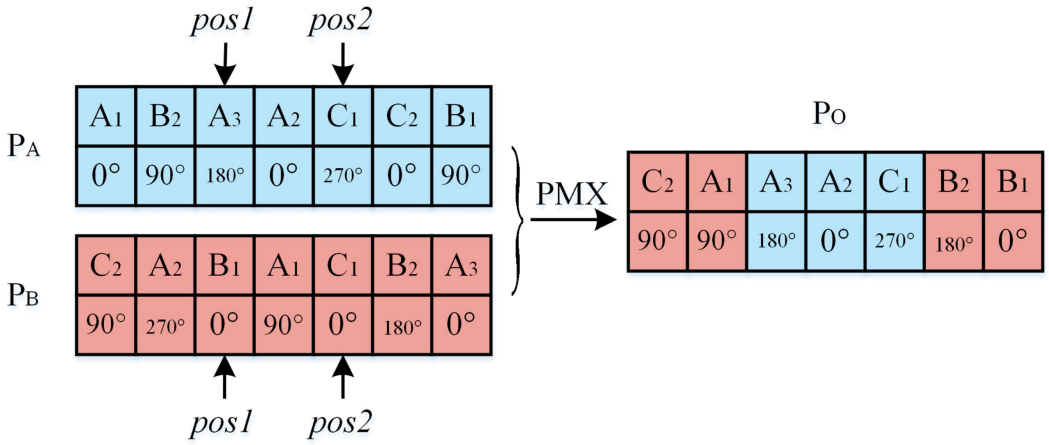


Figure 4. Partially mapped crossover operation.

4.3.3. Mutation Operator Design

Five mutation operators are designed. The first two are general-purpose perturbation operators, while the last three provide targeted improvement for the three optimization

objectives, respectively. All mutation operators act directly on the component sequence or the rotation vector, and the mutated chromosome is automatically decoded into a feasible layout scheme by the decoder.

(1) Adjacent Swap, M1

Randomly select a position pos_i in the encoded individual sequence, where $1 \leq pos_i < T$, and swap the two components at positions *pos_i* and $pos_i + 1$. This operation changes only the placement order of two adjacent components, with a relatively mild perturbation, and is suitable for local fine-tuning of a current high-quality sequence.

(2) Distant Swap, M2

Randomly select two positions pos_i and pos_j in the encoded individual sequence such that $|pos_i - pos_j| > |T/3|$, and swap the components at *pos_i* and *pos_j*. Since the two positions are relatively far apart in the sequence, the swapped components are likely to be assigned to different pallets after decoding, thereby producing a relatively strong structural perturbation and helping the algorithm escape local optima when search stagnates.

(3) Large-First Shift, M3

This mutation operator is based on the classical first-fit decreasing idea. In the process of assigning PC components to pallets, the earlier a large-area component is placed, the more likely it is to be compactly arranged on fewer pallets. The specific steps are as follows.

Step 1. In the latter half of the encoded individual sequence, find the component C^* with the largest area and record its current position as *pos_i*.

Step 2. Randomly select an insertion position *pos_j* in the first half of the sequence.

Step 3. Remove C^* from position *pos_i* and insert it at position *pos_j*, shifting the elements from *pos_j* to $pos_i - 1$ one position backward in sequence. By moving large-area components that appear later in the sequence to the earlier part of the sequence, they are processed earlier during decoding, thereby improving space utilization and reducing the number of pallets required.

(4) Balance-Guided Swap, M4

This mutation operator analyzes the currently decoded layout scheme, identifies pairs of pallets with imbalanced loads, and, in a targeted manner, adjusts the placement order of the relevant components within the sequence. The specific steps are as follows.

Step 1. Decode the current chromosome to obtain the pallet assignment information for each component.

Step 2. Calculate the carried area of each active pallet, and identify the heaviest-loaded pallet, denoted P_h, with the largest carried area, and the lightest-loaded pallet, denoted P_l, with the smallest carried area.

Step 3. Randomly select one component C_h from P_h and one component C_l from P_l, and find their positions *pos_h* and *pos_l* in the encoded individual sequence.

Step 4. Swap the components at positions *pos_h* and *pos_l*.

By swapping the placement order of components between the heaviest and lightest pallets in the sequence, the two components are likely to be reassigned to different pallets after decoding, achieving a peak-shaving and valley-filling balancing effect and thereby reducing fluctuations in pallet area utilization.

(5) Ergonomics-Guided Rotation, M5

This mutation operator acts directly on the rotation vector θ. By analyzing the global coordinates of embedded parts after decoding, it provides targeted improvement in the human factors ergonomics indicator. The specific steps are as follows.

Step 1. Decode the current chromosome and calculate the global coordinate *gy* of each embedded part of each component on the pallet.

Step 2. Randomly select a component C^* from all components that involve climbing operations. If no such component exists, randomly select any component.

Step 3. For C^*, try the other three rotation angles one by one while keeping the sequence unchanged, re-decode, and calculate the number of climbing operations caused by this component under each rotation orientation.

Step 4. Select the rotation angle with the fewest climbing operations to update the placement angle of component C^*. If multiple angles have the same number of climbing operations, randomly select one of them.

This operator does not change the component sequence but optimizes only the rotation orientation, so that embedded parts fall into the comfortable working zones on both sides of the pallet as much as possible, thereby reducing the frequency of climbing operations in a targeted manner.

Then, after crossover and mutation in each generation, NSGA-II performs non-dominated sorting, crowding-distance calculation, and elitist preservation on the merged parent and offspring populations, and selects offspring individuals via binary tournament selection. The iterative computation continues until the maximum number of iterations is reached.

4.4. Q-Learning Adaptive Parameter Control

4.4.1. Markov Decision Process Modeling

The intergenerational evolutionary process of NSGA-II is modeled as a Markov decision process. Before each generation t begins, the Q-Learning agent observes the macroscopic state s_t of the current population and selects an action a_t from the reproduction strategy library, that is, a complete set of crossover and mutation parameter configurations, for use in the reproduction operations of that generation. After the evolution of generation t ends, the agent obtains an immediate reward r_t based on the improvement of the Pareto front and observes the new population state s_{t+1}, thereby updating the policy value estimates in the Q-table.

4.4.2. State Space Design

To enable Q-Learning to perceive the dynamic changes in the population search process, this paper extracts three discretized feature variables at the population level to form a state space S of dimension 12.

Convergence feature $s_1 \in \{0, 1\}$. A sliding window width G_w is set. If within the most recent G_w generations, no new non-dominated individual appears in the first front of the Pareto solution set, the algorithm is considered to be in a temporary stagnation state, and $s_1 = 1$; otherwise, $s_1 = 0$. This feature reflects whether the search has encountered a bottleneck in approaching the Pareto front. When $s_1 = 1$, Q-Learning tends to select strategies with stronger perturbation to break the stagnation.

Diversity feature $s_2 \in \{0, 1\}$. Calculate the average crowding distance ($\overline{CD}$) of all individuals in the current population. If $\overline{CD}$ is below a preset threshold τ_{cd}, then $s_2 = 1$, indicating that individuals are too clustered in the objective space and diversity is insufficient; otherwise, $s_2 = 0$.

Objective bottleneck feature $s_3 \in \{0, 1, 2\}$. Identify which of the three objectives has the largest room for improvement in the current population. Specifically, for all individuals in the first front of the Pareto solution set, normalize the three objective values using the population's minimum and maximum values, and calculate the normalized mean for each objective dimension. The objective dimension with the largest normalized mean is the weak link of the current population. If f_1 is the bottleneck, $s_3 = 0$; if f_2 is the bottleneck, $s_3 = 1$; if f_3 is the bottleneck, $s_3 = 2$. This feature enables Q-Learning to learn an empirical mapping of which targeted strategy to invoke when a certain objective is weak. The combination of the above three features constitutes $|S| = 2 \times 2 \times 3 = 12$ discrete states.

4.4.3. Action Space Design

The action space consists of six predefined reproduction strategies, as presented in Table 1. Each strategy is a quadruple of (C_t, M_t, p_c, p_m), which respectively specify the crossover operator type, mutation operator type, crossover probability, and mutation probability used in t generation. The dimension of the Q-table is $|S| \times |A| = 12 \times 6 = 72$.

Table 1. Action space strategies.

Strategy	Crossover	p_c	Mutation	p_m	Design Intent
a_1	OX	0.9	M_1	0.1	High crossover rate for fine recombination of high-quality sequence segments; low mutation rate for local order fine-tuning; suited for the stable phase of the Pareto front.
a_2	PMX	0.9	M_2	0.4	MX deeply mixes parental genes; moderate-intensity distant swap broadens the search scope; suited for enhancing diversity when population convergence is insufficient.
a_3	OX	0.8	M_3	0.3	Preserves high-quality sequence segments; large-first shift targets the reduction in pallet count; activated when f_1 is identified as the bottleneck
a_4	OX	0.8	M_4	0.3	Preserves high-quality sequence segments; balance-guided swap targets the reduction in area variance; activated when f_2 is identified as the bottleneck.
a_5	PMX	0.8	M_5	0.3	PMX introduces new sequence combinations; ergonomics rotation targets the reduction in climbing operations; activated when f_3 is identified as the bottleneck.
a_6	PMX	0.6	M_2	0.4	Low crossover rate combined with high-intensity distant swap produces maximum structural disruption; suited for escaping deep stagnation of the Pareto front.

The specific probabilities for crossover (p_c) and mutation (p_m) in the reproduction strategy library were calibrated through preliminary sensitivity experiments. Strategies prioritizing exploitation (e.g., a_1) are assigned a high p_c and low p_m to preserve elite genetic segments, whereas exploration-driven strategies (e.g., a_6) utilize a higher p_m and lower p_c to disrupt local optima effectively.

The design of these strategies reflects three levels of consideration. a_1 and a_6 correspond to pure exploitation and pure exploration, respectively, representing the two ends of the search spectrum and regulating the overall search pace. The three objective-oriented strategies a_3, a_4, and a_5 correspond to the three bottleneck types of the state feature s_3, respectively. a_2, as a general exploration strategy, is applicable under various intermediate states.

4.4.4. Reward Function Design

After each generation of evolution, Q-Learning needs to evaluate the actual effect of the selected strategy. This paper uses the degree of Pareto front improvement and the competitiveness of offspring as a two-dimensional reward signal. Let the first Pareto front of generation t be $F_1^{(t)}$ and that of generation $t-1$ be $F_1^{(t-1)}$. Define the front update number n_{new} as the number of newly appeared non-dominated individuals in $F_1^{(t)}$, that is, individuals present in the current front but not in the previous front. Define the offspring survival rate ρ_s as the proportion of offspring individuals generated in the current generation that successfully enter the next population.

The reward function is assigned according to the stepwise rule in Equation (21). This reward function indicates that strategies capable of advancing the Pareto front receive the highest positive incentive. Strategies that do not break through the front but have a relatively high offspring survival rate indicate that the strategy direction is basically correct and are given mild encouragement. If almost all offspring are eliminated by environmental

selection, it indicates that the strategy does not match the current population state, and a moderate penalty is applied to guide Q-Learning in adjusting subsequent choices.

$$r_t = \begin{cases} +1.0, & \text{if } n_{\text{new}} \geq 2 \text{ (strategy significantly advances the Pareto front)} \\ +0.5, & \text{if } n_{\text{new}} = 1 \text{ (strategy effectively advances the Pareto front)} \\ +0.1, & \text{if } n_{\text{new}} = 0 \text{ and } \rho_s \geq 0.2 \text{ (offspring are competitive but fail to break through the front)} \\ -0.3, & \text{if } n_{\text{new}} = 0 \text{ and } \rho_s < 0.2 \text{ (strategy fails to produce effective offspring)} \end{cases} \tag{21}$$

4.4.5. Q-Value Update and Exploration-Exploitation Strategy

At the end of each generation, upon receiving the reward r_t, Q-Learning updates the Q-table according to the Bellman Equation (22) [50]. Here, α is the learning rate, γ is the discount factor, and s_{t+1} is the observed state of the new generation population.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot [r_t + \gamma \cdot \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)] \tag{22}$$

During strategy selection, the ε-greedy strategy, as shown in Equation (23), is employed to balance exploration and exploitation. The exploration rate decays linearly with the evolutionary generation. In the early stage of evolution, a relatively high ε helps the agent try various strategies broadly to accumulate experience. As generations progress, ε gradually decreases, and the agent shifts to exploiting the learned Q-values, focusing on selecting the most effective strategy for the current state. ε_{max} and ε_{min} denote the initial upper and lower bounds of the exploration rate, respectively, and G_{max} is the maximum number of evolutionary generations.

$$\varepsilon_t = \varepsilon_{\max} - (\varepsilon_{\max} - \varepsilon_{\min}) \cdot \frac{t}{G_{\max}} \tag{23}$$

4.5. Overall Algorithm Execution Workflow

Integrating the above modules, the QL-NSGA-II algorithm consists of two interacting parts: the NSGA-II evolutionary environment and the online Q-Learning parameter controller. The complete execution workflow of the algorithm is illustrated in Figure 5 and detailed as follows.

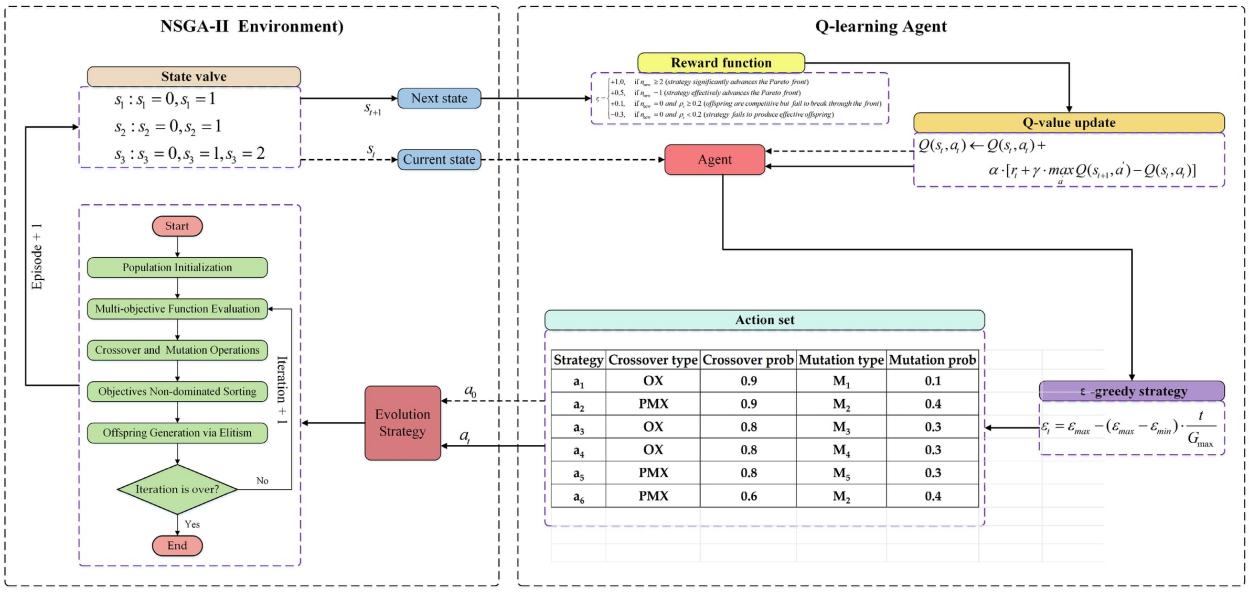


Figure 5. Flowchart of the QL-NSGA-II.

Step 1 (Initialize environment). Set the population size N_{pop} and the maximum number of evolutionary generations (maximum steps per episode) G_{max}.

Step 2 (Set training parameters). Set parameters such as the maximum number of episodes, sliding window width G_w, learning rate α, discount factor γ, maximum ε_{max}, and minimum ε_{min}.

Step 3 (Initialize Q-table). Initialize the Q-table as a zero matrix.

Step 4. For each episode, initialize the population state.

Step 5. NSGA-II evolutionary generation iterations/steps within each episode.

Substep 5.1 (State observation). Before generation t begins, extract the three-dimensional macroscopic state $s_t = (s_1, s_2, s_3)$ of the population Pop_t.

Substep 5.2 (Strategy selection). The Q-Learning agent uses the ϵ_t-greedy strategy to select a strategy $a_t = (C_t, M_t, p_c, p_m)$ from the reproduction strategy library.

Substep 5.3 (Reproduction operation). Select parent individuals from Pop_t via binary tournament, apply the crossover operator C_t to the component sequences of paired parents with probability p_t, and apply the mutation operator M_t to each offspring with probability p_m. Decode all offspring chromosomes into complete layout schemes and evaluate objective values to obtain the offspring population O_t.

Substep 5.4 (Environmental selection). Merge Pop_t and O_t, perform non-dominated sorting and crowding distance calculation, and select N_{pop} individuals according to the rule of front-level priority followed by crowding-distance priority to form Pop_{t+1}.

Substep 5.5 (Reward evaluation and Q-table update). Count the front update number n_{new} and offspring survival rate ρ_s, and calculate the immediate reward r_t. Observe the new population state s_{t+1} and update the Q-table according to the Bellman equation.

Substep 5.6 (State judgment). If s_{t+1} is a terminal state, proceed to Step 6; otherwise, return to Substep 5.1.

Step 6 (Episode training termination). Proceed to Step 4 to train the next episode.

Step 7 (Training termination). Terminate training when the maximum number of training episodes is reached.

Step 8 (Extract optimal strategy). Extract the maximum value from the Q-table to obtain the approximately optimal strategy.

Step 9 (Apply optimal strategy). Apply the optimal Q-table strategy and output the final set of Pareto-optimal pallet layout schemes.

5. Experimental Study and Analysis

5.1. Experimental Setup

The dimensions of the PC components and the location of the embedded parts in the test case were based on actual order data from a prefabricated component factory in Zigong City, Sichuan Province, China. The effective working surface dimensions of the pallet are uniformly set to $L \times W = 10$ m $\times$ 4 m, and the minimum production process clearance is $d = 0.3$ m. The basic information of the test cases is detailed in Table 2.

Table 2. PC component production order information.

PC	Order Demand	PC Size (mm)		Embedded Part Positions
		Length	Width	Coordinates (x, y)
1	2	1970	1985	-
2	4	2300	1590	(1675, 1490)
3	5	2310	2120	(135, 980) (1290, 700) (1965, 1935) (2115, 1690)
4	3	2310	2120	-

Table 2. *Cont.*

PC	Order Demand	PC Size (mm)		Embedded Part Positions
		Length	Width	Coordinates (x, y)
5	4	2310	2120	(2240, 940)
6	5	2310	2120	(1450, 940)
7	2	2310	2120	(1835, 940) (2200, 955)
8	1	2310	2120	(280, 345) (1650, 510) (2235, 1095)
9	1	2310	2120	(815, 1545) (1045, 1125) (1385, 940) (2025, 570)
10	3	2420	2360	(480, 190) (945, 195) (1760, 540)
11	2	2420	2360	(665, 1100) (855, 1210) (1500, 560) (1575, 1155) (1590, 395) (1595, 1325) (1590, 1945) (1935, 1237)
12	3	2420	1920	(835, 770) (1880, 770)
13	3	2420	2385	(835, 595) (1880, 595)
14	3	2590	1630	(610, 110)
15	1	2670	2350	-
16	1	2670	2350	(520, 601) (710, 1035) (713, 710) (1380, 1020)
17	5	2670	2350	(760, 395) (765, 75) (1140, 2235) (1965, 1130) (2440, 1800)
18	1	2670	2350	(1500, 1270) (1540, 930)
19	2	2670	2350	(1130, 1600)
20	1	2670	1910	-
21	1	2670	2260	(1450, 1660)
22	1	2670	2310	(750, 390) (730, 1020) (2470, 600)
23	1	2670	1985	-
24	4	2670	2235	(2040, 325)
25	5	2670	2160	(635, 775)
26	4	2670	1780	(2560, 1670)
27	3	2770	2350	(980, 210) (1225, 210) (1380, 1065) (1790, 210) (2135, 210) (2660, 1030)
28	1	2770	2110	(1280, 675)
29	2	3020	2420	(200, 1760) (400, 920) (1190, 1760) (1190, 920) (2495, 1000) (2885, 475)
30	4	3020	2420	(435, 120)(435, 1880)(1390, 1880)
31	2	3070	2420	(461, 621) (478, 959) (1320, 1760) (1857, 743) (2475, 2314) (2507, 743) (1028, 470)
32	5	3370	2360	(1695, 1055) (1905, 1085) (1915, 485)
33	5	3420	2985	(1520, 1935)
34	4	3420	2000	-
35	4	3420	2420	(2520, 1255) (2715, 2325) (2780, 405) (3055, 1435)
36	3	3510	3020	(900, 82) (455, 900) (1000, 805) (1570, 1630) (2360, 1425) (2910, 1950) (3150, 1105)
37	4	3785	2120	(290, 420) (450, 1405) (485, 215) (1160, 1985) (1603, 1403) (1230, 1501) (2090, 700)
38	1	3785	2120	(135, 920) (935, 675) (3030, 925)
39	1	3785	2120	(255, 900) (805, 700) (835, 525) (835, 210) (3605, 700)
40	4	3785	2120	(1150, 955) (2475, 925) (3670, 1220)
41	4	3785	2420	(210, 185) (965, 185) (965, 385) (2015, 1880) (2015, 820)
42	2	4020	2330	(295, 1775) (1195, 575) (3348, 1773)
43	5	4020	2110	(670, 630) (1740, 580) (3725, 630)
44	3	4320	2420	(625, 957) (945, 590) (1305, 740) (1330, 1030) (2380, 1030) (3305, 1675) (3460, 1760) (4205, 957)
45	2	4320	2420	(95, 1030) (285, 1030) (1015, 778) (2075, 1300) (3415, 778)

This batch of orders covers 45 types of rectangular precast components with different specifications. The demand per order for each type ranges from 1 to 5 units, for a total of 127 components. In terms of component dimensions, the length ranges from 1970 to 4320 mm, and the width ranges from 1590 to 3020 mm. Considerable dimensional differences

exist among component types, including small panels such as type 1 (approximately 3.91 m^2) and large panels such as type 44 (approximately 10.45 m^2). Regarding embedded parts, 6 of the 45 component types, namely types 1, 4, 15, 20, 23, and 34, do not contain embedded parts, while the remaining 39 types each contain a varying number of embedded parts, ranging from 1 to 8. The installation positions of each wire box are listed in the table in local coordinates (x, y). The embedded part layout of some components is relatively complex. For example, a single unit of type 11 contains 8 parts, types 31 and 37 each contain 7, and types 27 and 29 each contain 6. This dimensional and structural heterogeneity substantially increases the combinatorial complexity of the layout problem. Although the test cases are derived from a single production order, this specific batch represents a highly complex and heterogeneous scenario (45 distinct PC types with extreme variations in dimensions and embedded part layouts), making it a rigorous benchmark for evaluating layout optimization algorithms.

This paper divides the order data in Table 2 into three test case levels, namely small, medium, and large, and combines them with three different working depth modes to construct $3 \times 3 = 9$ experimental configurations. Specifically, regarding case-scale division, this paper selects different numbers of component combinations based on the order sequence. Components of types 1 to 8, totaling 26 PC components (P26), form the small-scale test case. Components of types 1 to 25, totaling 64 PC components (P64), form the medium-scale test case. The full order set, totaling 127 PC components (P127), forms the large-scale test case.

Meanwhile, for the setting of the working depth parameter δ, this paper refers to the comfortable arm reach of a male worker of average height when standing at the pallet edge and divides it into three engineering constraint modes [51]: 450 mm for the strict mode, 650 mm for the standard mode, and 800 mm for the relaxed mode. The selection of working depths (450 mm, 650 mm, and 800 mm) is grounded in both standard anthropometric data (e.g., ISO 11228 and Chinese adult human body dimensions) and empirical field measurements obtained from the PC component manufacturing enterprise in Zigong City. Specifically, 450 mm corresponds to the strict primary reach zone, 650 mm aligns with the standard functional reach of an average male worker, and 800 mm represents a relaxed reach requiring posture extension or the use of specific hand tools.

5.2. Comparative Algorithms, Parameter Settings, and Performance Metrics

To systematically evaluate the performance advantages of the proposed QL-NSGA-II algorithm, three representative algorithms are selected as benchmarks for comparison. The first is the standard NSGA-II algorithm (S-NSGA-II), which employs fixed-order crossover (OX) with a crossover probability of 0.8 and adjacent-swap mutation (M1) with a mutation probability of 0.2, representing a conventional solution strategy without adaptive mechanisms. The second is the random operator selection NSGA-II algorithm (R-NSGA-II), which uses the same crossover and mutation operator library as QL-NSGA-II but randomly selects reproduction strategies with equal probability in each generation, aiming to isolate and verify the core contribution of the Q-Learning intelligent scheduling mechanism to performance improvement. The third is the multi-objective evolutionary algorithm based on decomposition, named MOEA/D, which uses the Tchebycheff decomposition method with a neighborhood size of 20 and represents another classical multi-objective optimization paradigm.

In terms of parameter configuration, to ensure fairness in the comparison, the population sizes of the four algorithms across the three case scales are set to 50, 100, and 200, respectively, and the maximum number of evolutionary generations is set to 200, 300, and 400, respectively. For the reinforcement learning control module of QL-NSGA-II, the learning rate of Q-Learning is set to 0.1, the discount factor is set to 0.9, the exploration-

exploitation balance parameter ε decays linearly from an initial value of 0.8 to 0.05 as generations proceed, and the sliding window width for stagnation detection used in feature state perception is set to 10 generations.

To ensure a strictly fair comparison, all four baseline algorithms are allocated the exact same computational evaluation budget ($N_{pop} \times G_{max}$) and shared identical stopping criteria. Furthermore, while R-NSGA-II utilizes the same diversified operator library as QL-NSGA-II, its operator selection is entirely stochastic, isolating the performance contribution directly attributable to the Q-learning state-action mapping.

For performance evaluation, four widely used multi-objective optimization metrics are selected to quantitatively analyze the Pareto fronts produced by each algorithm. The first is Hypervolume (HV), which can simultaneously measure both the convergence and the breadth of a solution set's distribution. A larger value indicates higher overall solution quality of the algorithm. The reference point used in the calculation is set to the maximum value of each objective across all runs of all algorithms, multiplied by a safety factor of 1.5. The second is Inverted Generational Distance (IGD), which measures the degree to which the front obtained by an algorithm approximates and covers the true Pareto front. A smaller IGD value indicates better convergence. Since this problem is a complex discrete combinatorial optimization problem and the true Pareto front is unknown, the union of non-dominated solutions generated by all algorithms across all independent runs is used as an approximate reference front. In addition, the Spacing (SP) metric is introduced to evaluate the uniformity of solution distribution on the Pareto front. A smaller SP value indicates a smoother distribution of non-dominated solutions in the objective space. Finally, the number of non-dominated solutions in the first front output by each algorithm, denoted as $|F_1|$, is used as an intuitive auxiliary evaluation criterion. Given similar convergence quality, a larger number of non-dominated solutions can provide factory managers with a richer decision space for PC production scheduling.

5.3. Comparative Analysis of Algorithm Performance

5.3.1. Global Comparison of HV Metrics

Table 3 summarizes the mean HV values of the four algorithms under nine experimental configurations. Overall, QL-NSGA-II achieves the highest HV values in all nine configurations, but the performance gaps among the four algorithms show a differentiation pattern closely related to problem scale. In P26, the HV performance of the four algorithms is quite close. Taking $\delta = 650$ mm as an example, the gap between the best QL-NSGA-II at 0.7923 and the lowest MOEA/D at 0.7834 is only 1.1%, and under the three δ modes, this gap does not exceed 1.5%. This indicates that when the number of components is limited and the search space scale is manageable, the optimization capabilities of various algorithms are relatively balanced. The fixed-parameter strategy of standard NSGA-II can already achieve near-optimal front quality, and the additional benefit of the Q-Learning adaptive mechanism is not yet obvious.

However, as the problem scale increases, performance differentiation among the algorithms gradually emerges. In P64, the HV advantage of QL-NSGA-II over S-NSGA-II expands to 3.6% to 3.7%, and its advantage over R-NSGA-II is about 2.4% to 2.7%. In P127, this trend of differentiation intensifies further. The mean HV of QL-NSGA-II stabilizes around 0.70, while that of S-NSGA-II drops to about 0.61, and the gap between them jumps to 12.1% to 14.2%. Even R-NSGA-II, which has the same operator library, has an HV about 7.4% to 8.1% lower than that of QL-NSGA-II. This result indicates that when the permutation and rotation combinations of 127 components yield a relatively large search space, the static balance between exploration and exploitation of the fixed-parameter strategy becomes increasingly difficult to maintain as the population state evolves. In contrast,

Q-Learning can dynamically adjust reproduction strategies based on real-time feedback on convergence pressure, diversity level, and objective bottlenecks, thereby unlocking greater search potential in high-dimensional, complex problems.

Table 3. Comparison of mean HV metrics among four algorithms on three test cases.

Case	δ (mm)	S-NSGA-II	R-NSGA-II	MOEA/D	QL-NSGA-II
P26	450	0.7845	0.7867	0.7812	0.7912
	650	0.7889	0.7856	0.7834	0.7923
	800	0.7823	0.7878	0.7806	0.7901
P64	450	0.7145	0.7234	0.7178	0.7412
	650	0.7212	0.7289	0.7234	0.7478
	800	0.7123	0.7198	0.7156	0.7389
P127	450	0.6178	0.6512	0.6345	0.7034
	650	0.6089	0.6434	0.6278	0.6956
	800	0.6234	0.6567	0.6389	0.7089

5.3.2. Detailed Analysis of All Metrics for a Representative Case

Among the nine experimental configurations above, P127, with a working range of $\delta = 650$ mm, is selected as a representative configuration for a comprehensive comparison of the four metrics. On the one hand, the large-scale case contains all 127 components, and the complexity of its search space can sufficiently amplify the performance differences among algorithms. On the other hand, $\delta = 650$ mm corresponds to typical factory operating conditions, and its results serve as a relatively high reference for engineering practice. The means and standard deviations of the metrics for the four algorithms under this configuration are shown in Table 4.

Table 4. Comparison of all metrics for all algorithms on P127 at working range $\delta = 650$ mm.

Metric	S-NSGA-II	R-NSGA-II	MOEA/D	QL-NSGA-II
HV	0.6089 ± 0.031	0.6434 ± 0.025	0.6278 ± 0.028	0.6956 ± 0.020
IGD	0.1345 ± 0.016	0.1056 ± 0.013	0.1178 ± 0.015	0.0789 ± 0.009
SP	0.0478 ± 0.008	0.0389 ± 0.006	0.0434 ± 0.007	0.0312 ± 0.004
$\lvert F_1 \rvert$	35.8 ± 4.5	41.6 ± 3.4	17.2 ± 4.8	53.4 ± 3.1

As shown in Table 4, QL-NSGA-II achieves the best results across all four metrics. In terms of convergence, the mean HV of QL-NSGA-II is 0.6956, which is 14.2% higher than 0.6089 of S-NSGA-II and 8.1% higher than 0.6434 of R-NSGA-II. Its mean IGD is 0.0789, which is 41.3% and 25.3% lower than those of S-NSGA-II and R-NSGA-II, respectively, indicating that the proposed algorithm can more closely approximate the reference Pareto front. In terms of distribution uniformity, the SP value of QL-NSGA-II is 0.0312, which is 34.7% lower than 0.0478 of S-NSGA-II. This indicates that the non-dominated solutions obtained by QL-NSGA-II are more evenly distributed in the objective space and can more fully characterize the geometric shape of the Pareto front. In terms of solution set size, QL-NSGA-II outputs an average of 53.4 non-dominated solutions, which is 49.16% and 28.37% more than S-NSGA-II and R-NSGA-II, respectively, providing decision-makers with richer trade-off options.

It is worth noting that QL-NSGA-II not only has better mean values for each metric but also has the smallest standard deviations across the four metrics. This indicates that adaptive strategy control in Q-Learning improves solution quality while enhancing the algorithm's stability and consistency across multiple independent runs. In contrast, because S-NSGA-II relies on fixed crossover and mutation parameters throughout, its results are more sensitive to the randomness of the initial population, leading to relatively large

standard deviations. Although R-NSGA-II generally outperforms S-NSGA-II in various metrics due to its diverse operator library, a notable gap remains between R-NSGA-II and QL-NSGA-II. This comparison directly demonstrates that the core source of performance improvement is not the richness of the operator library itself, but rather the intelligent scheduling of operator invocation timing by Q-Learning. MOEA/D performs relatively weakly in terms of solution set size. This may be related to the difficulty of its framework, which is based on weight vector decomposition, in covering certain regions when facing non-convex and discontinuous Pareto fronts among objectives.

5.3.3. Statistical Significance Testing and Effect Size Analysis

Given the stochastic nature of evolutionary algorithms, Cohen's effect size (ES) is employed alongside the non-parametric Wilcoxon test to quantify the standardized magnitude of algorithmic improvement. This paper conducts the Wilcoxon rank-sum test at a significance level of 0.05 on the 30 independent run results of QL-NSGA-II and the three comparison algorithms in terms of the HV and ES metrics to determine the practical magnitude of the differences. The effect size is calculated as $ES = \left|\overline{HV_1} - \overline{HV_2}\right| / PSD$, where $\overline{HV_1}$ and $\overline{HV_2}$ are the mean HV values of the two algorithms, and PSD is the pooled standard deviation. According to the general criteria proposed by Cohen [52], $ES < 0.2$ indicates a negligible effect, $0.2 \leq ES < 0.5$ a small effect, $0.5 \leq ES < 0.8$ a medium effect, and $ES \geq 0.8$ a large effect.

The test results are shown in Table 5, from which three levels of conclusions can be drawn. First, in P26, none of the differences between the comparison groups are statistically significant ($p > 0.05$), and Cohen's ES values mainly fall within the small-to-medium effect range (0.14 to 0.59). This indicates that when the number of components is limited and the search space is relatively small, conventional algorithms with fixed parameters can already effectively approach the Pareto front, and the additional benefit of the Q-Learning adaptive mechanism is not yet significant. Second, as the problem scale increases to P64, the differences between QL-NSGA-II and the three comparison algorithms are all statistically significant ($p < 0.05$), and Cohen's ES values all exceed the large effect threshold of 0.8. This indicates that as search difficulty increases, the intelligent scheduling mechanism of Q-Learning, which dynamically switches operators based on population states, begins to yield substantial performance gains. Third, in the complex large-scale case P127, all comparison items exhibit strong statistical significance ($p < 0.001$) and very large effect sizes ($ES > 2.0$). Overall, Cohen's ES shows a notable monotonic increase as the problem scale expands. For example, compared with S-NSGA-II, the ES value increases from 0.20 to 3.21. This provides strong evidence that the Q-Learning adaptive control mechanism has excellent optimization performance and good scalability when dealing with the expansion of high-dimensional search spaces.

Table 5. Wilcoxon test p-values and Cohen's ES on HV for QL-NSGA-II and comparison algorithms.

Case	δ (mm)	vs. S-NSGA-II	vs. R-NSGA-II	vs. MOEA/D
P26	450	0.156/0.39	0.312/0.26	0.068/0.59
	650	0.423/0.20	0.189/0.39	0.092/0.52
	800	0.098/0.46	0.567/0.14	0.078/0.56
P64	450	0.003/1.18	0.042/0.82	0.006/1.08
	650	0.002/1.23	0.023/0.93	0.004/1.14
	800	0.004/1.12	0.035/0.87	0.007/1.02
P127	450	<0.001/3.06	<0.001/2.09	<0.001/2.55
	650	<0.001/3.21	<0.001/2.15	<0.001/2.63
	800	<0.001/2.98	<0.001/2.03	<0.001/2.47

5.4. Sensitivity Analysis of Working Range Parameters

In the pallet layout planning for composite slabs, the working range δ directly determines the comfortable reach of workers standing along the long side of the pallet to install side molds and embedded parts without climbing onto the pallet. A smaller δ leaves a larger blind zone in the central area of the pallet, and components are likely to fall into this blind zone, thereby triggering additional climbing operations. Conversely, a larger δ provides wider coverage of the working zone. Therefore, the value of δ not only reflects the actual human factors ergonomic constraints in the factory but also directly determines the difficulty of optimizing the third objective. Table 6 presents the Pareto front characteristics of QL-NSGA-II under three δ modes.

Table 6. Pareto front characteristics of QL-NSGA-II in P127 under different δ modes.

Metric	δ = 450 mm	δ = 650 mm	δ = 800 mm
HV	0.7034 ± 0.024	0.6956 ± 0.020	0.7089 ± 0.019
$\lvert F_1 \rvert$	63.2 ± 3.5	53.4 ± 3.1	64.6 ± 2.8
f_1^*	40	40	40
f_2^*	1.37	1.44	1.38
f_3^*	298	267	220

Note: f^* denotes the minimum value of the objective.

It should be noted that the same δ parameter sensitivity test was also conducted on the small-scale case P26 and the medium-scale case P64. The experimental results show an evolutionary pattern highly consistent with that of P127. However, because the search spaces in the small and medium cases are limited and the conflicts among objectives are relatively weak, their sensitivity to changes in δ is less pronounced than in the large-scale scenario. To avoid redundancy and highlight the essence of the problem, this section focuses on the large-scale case P127, which represents the most challenging engineering environment, for sensitivity analysis.

From the statistical data in Table 6, the following observations can be made. First, relaxing the working range δ effectively releases the constraints of the solution space, thereby improving the overall performance of the algorithm. As the working range increases from 450 mm to 800 mm, the highest mean HV value reaches 0.7089. Although the HV metric shows a slight fluctuation at 650 mm, the overall trend still indicates that moderately relaxing the ergonomic constraint boundary helps the algorithm search for a Pareto front with higher overall quality and broader coverage. Second, across the three modes, the optimal number of pallets is 40, and the optimal pallet area utilization fluctuates only within a small range. This indicates that the proposed algorithm is highly robust in the macroscopic space utilization dimension and can adaptively absorb disturbances arising from changes in ergonomic constraints without sacrificing core economic indicators. Third, as δ gradually increases from 450 mm to 800 mm, the number of embedded parts requiring climbing operations under the optimal state drops from 298 to 220, a reduction of 26.17%. This data intuitively confirms the engineering value of human factors ergonomics optimization. It is important to note that this reduction represents the theoretical benefit of relaxing ergonomic constraints rather than a direct algorithmic improvement.

5.5. Analysis of Q-Learning Behavior and Computational Efficiency

5.5.1. Phased Evolution of Strategy Selection

To reveal the computational logic of Q-Learning in the high-dimensional complex pallet layout space, this section takes the statistical average of 30 independent runs of P127 at δ = 650 mm as the basis and divides the complete 400-generation evolutionary process

into three stages: early, middle, and late. The average frequency with which each of the six reproduction strategies is selected by the agent at each stage is shown in Figure 6.

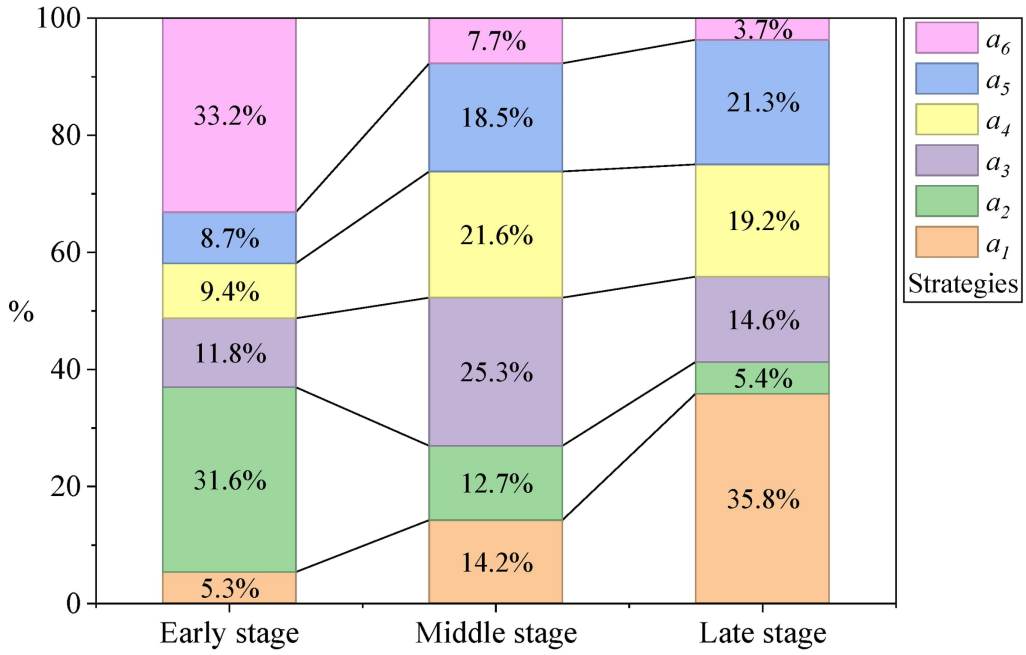


Figure 6. Stacked chart of the phased evolution of strategy selection.

From the shifting color-block areas in Figure 6, it is clear that in the early stage of evolution, Q-Learning is dominated by global exploration and escaping local optima. The exploration-oriented strategy a_2 and the strong perturbation strategy a_6 are dominant, with a combined average proportion of 64.8%. This is because the theoretical layout space for 127 components is considerably large, whereas the initial population covers a relatively narrow region in the objective space. Q-Learning readily captures the diversity pressure and adopts proactive, high-probability mutation and position-shuffling strategies to drive the population to disperse rapidly across the broad solution space. In the middle stage of evolution, the color block of the global exploration strategy shrinks rapidly, and the three objective-oriented strategies a_3, a_4 and a_5 gain dominance, with a combined proportion rising to 65.4%. Among them, a_3 has the highest proportion at 25.3%, indicating that, at this stage, how to place 127 components as compactly as possible on the minimum number of pallets is the primary bottleneck. As iterations proceed, the invocation frequencies of a_4 and a_5 also increase notably, showing a stepped improvement path that first addresses space utilization and then optimizes balance and human ergonomics. In the late stage of evolution, Q-Learning greatly increases the invocation frequency of a_1, expanding it to 35.8%. This strategy invokes OX with high probability to preserve as many well-formed, high-quality layout segments as possible in the parents. It strictly reduces the probability of mutation, thereby consolidating the existing Pareto solution set within the local neighborhood.

5.5.2. Convergence and Computational Efficiency Analysis

Figure 7 shows the convergence evolution curves of HV and IGD with evolutionary generations for the four algorithms on the P127 case under the $\delta = 650$ mm mode. In the early evolutionary stage (approximately generations 0 to 50), the HV curves of all four algorithms show a similar rapid upward trend, while the IGD curves decline sharply. This indicates that when the search space has not yet been fully explored, the influence of different parameter control strategies on optimization speed is relatively limited. After generation

50, the performance trajectories of the algorithms clearly differentiate. The HV curve of S-NSGA-II levels off first, and its IGD decline rate slows notably, suggesting that combining fixed operators with fixed probabilities is more likely to lead to search stagnation in the complex search space of 127 components. The convergence trajectories of R-NSGA-II and MOEA/D remain relatively close throughout the evolutionary process, and both outperform S-NSGA-II, with R-NSGA-II showing a slight advantage. This suggests that introducing diversified random operators makes a positive contribution to search performance.

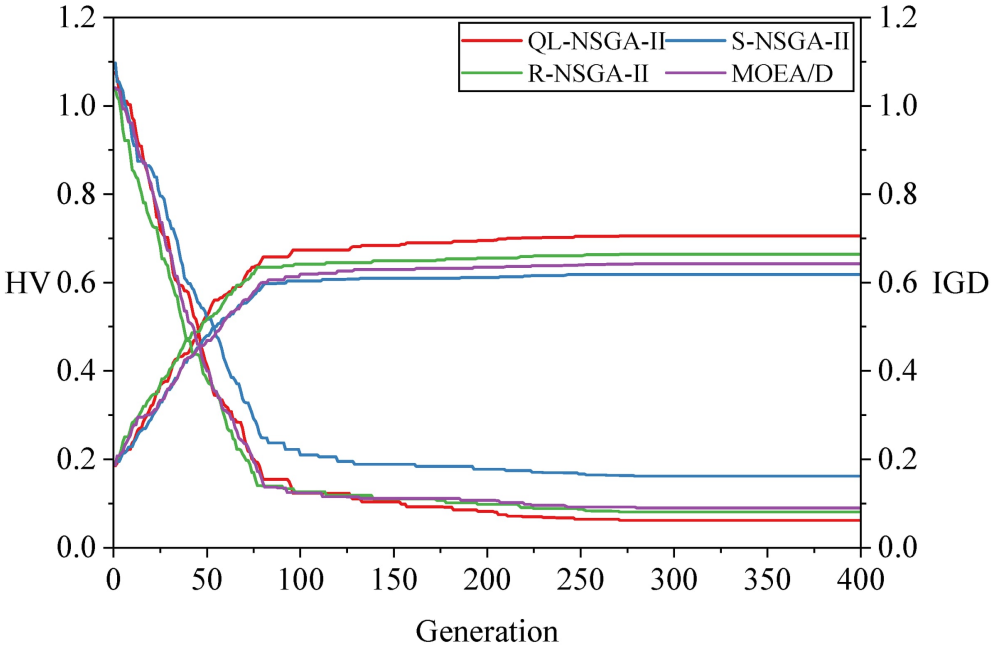


Figure 7. Convergence evolution curves of HV and IGD for P127 at working range $\delta = 650$ mm.

Notably, after about generation 80 to 100, the HV curve of QL-NSGA-II opens a discernible gap with the other three algorithms and maintains a relatively stable upward slope from generation 100 to 250, while its corresponding IGD curve continues to approach lower levels. This aligns with Q-Learning's behavioral pattern: it adaptively schedules objective-oriented mutation strategies based on population–state feedback in the middle evolutionary stage and switches to fine exploitation strategies in the later stage. Ultimately, the mean HV curve of QL-NSGA-II converges to 0.6956, and its mean IGD curve converges to 0.0789, achieving the highest final-state HV and the lowest final-state IGD among the four algorithms, respectively (consistent with the statistical results in Table 4). In contrast, S-NSGA-II's HV converges only to 0.6089, and its IGD remains at 0.1345, exhibiting a pronounced performance gap relative to QL-NSGA-II.

To evaluate the computational overhead of the proposed algorithm under high-load conditions, Table 7 summarizes the average computation time (in seconds) over 30 runs for the four algorithms on the large-scale case P127, with a maximum number of generations Gmax = 400, across different working ranges. As shown in Table 7, MOEA/D and S-NSGA-II are the fastest because they do not require state perception or strategy evaluation, taking approximately 780 s and 795 s, respectively, under the $\delta = 650$ mm mode. In comparison, QL-NSGA-II takes the longest time, about 874 s. This is mainly due to the necessary computational cost of extracting state features such as diversity and objective bottlenecks, calculating the offspring survival rate, and updating the Q-table in each generation, which increases the total computation time by only about 10%, approximately 1.1 times that of S-NSGA-II. This indicates that the introduction of the Q-Learning mechanism does not lead to exponential growth in complexity.

Table 7. Comparison of average computation time of algorithms on P127 (unit: s).

δ (mm)	S-NSGA-II	R-NSGA-II	MOEA/D	QL-NSGA-II
450	798.5	851.3	782.4	891.5
650	795.2	846.7	779.6	873.8
800	791.4	841.5	776.8	862.4

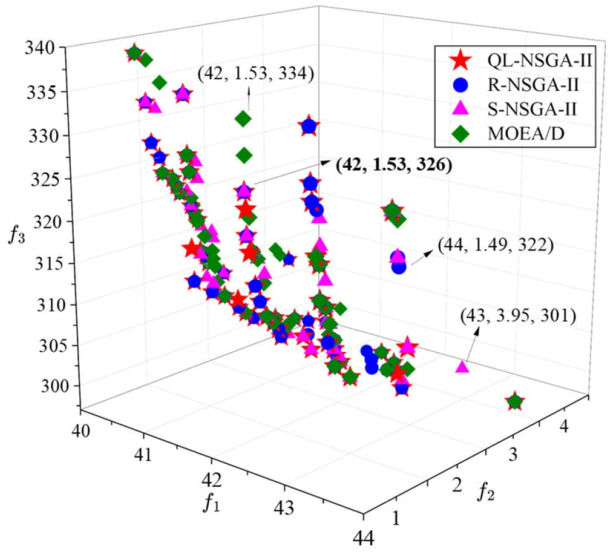
Although QL-NSGA-II is somewhat less efficient than traditional algorithms, under the pallet layout for 127 components, the total runtime for a complete run remains within 15 min (about 892 s). For macro-level production scheduling in actual prefabricated factories, the substantial benefits it provides in reducing pallet costs and mitigating elevated work risks make the computational delay acceptable, giving the algorithm good practical value.

5.5.3. Representative Pallet Layout Schemes

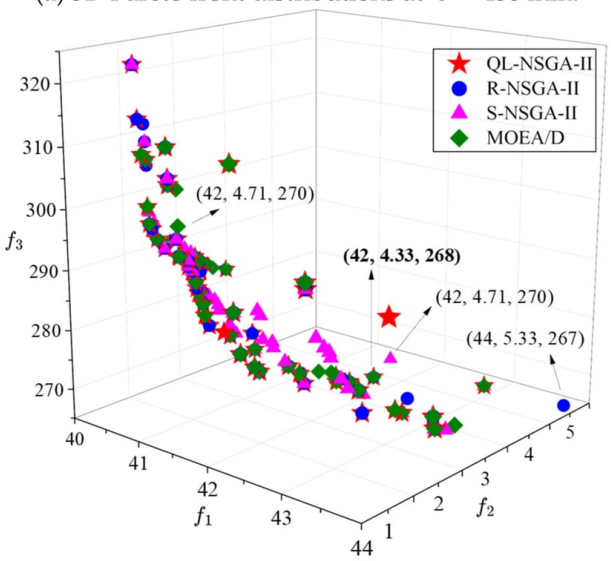
Figure 8 visually presents the quantitative distribution and evolution patterns of the Pareto-optimal solution sets obtained by the four algorithms for P127 under three working ranges. In terms of macroscopic spatial form and solution set size, QL-NSGA-II obtains 64, 54, and 65 non-dominated solutions, respectively, maintaining the highest or jointly highest solution set capacity among the comparison algorithms, whose solution counts mostly fall in the range of 47 to 57. As the working range gradually relaxes, the solution sets for the number of pallets (40 to 44) and the variance of the loaded area across all activated pallets (1.37 to 5.33) remain relatively stable. In contrast, the number of climbing operations shows a notable stepwise downward shift. Under the 450 mm mode, f_3 is concentrated in the interval (298, 338). Under the 650 mm mode, f_3 shifts to (267, 321), with the minimum decreasing from 298 to 267, a reduction of about 10.4%. Under the 800 mm mode, f_3 further shrinks to (220, 278), with the minimum reaching 220, a cumulative reduction of 78 climbing operations compared with the 450 mm mode, corresponding to an overall reduction of 26.2%.

From the perspective of quantitative dominance among individual solutions, the QL-NSGA-II solutions marked in bold black in the figure dominate the non-bold solutions from the comparison algorithms, and the remaining non-bold solutions are likewise dominated by the QL-NSGA-II. At $\delta = 450$ mm, compared with the MOEA/D solution (42, 1.53, 334), the bold solution (42, 1.53, 326) reduces the number of climbing operations by 8 while keeping f_1 and f_2 the same. Meanwhile, the solutions (44, 1.49, 322) and (43, 3.95, 301) are strictly dominated by the QL-NSGA-II solutions (43, 1.47, 321) and (42, 3.77, 301), respectively. At $\delta = 650$ mm, compared with the S-NSGA-II solution (42, 4.71, 270), the bold solution (42, 4.33, 268) reduces area variance from 4.71 to 4.33 with the same number of pallets and reduces the number of climbing operations from 270 to 268. The edge solution (44, 5.33, 267) is suppressed by the QL-NSGA-II solution (43, 5.24, 267) in two dimensions. Under $\delta = 800$ mm, the bold solution (43, 1.41, 257) achieves comprehensive dominance over the R-NSGA-II solution (44, 1.46, 266) across all three objectives. The MOEA/D solution (42, 1.44, 278) and the S-NSGA-II solution (42, 4.4, 223) are also dominated by the QL-NSGA-II solutions (42, 1.43, 270) and (42, 3.53, 223), respectively. The bolded solutions in the figures are relatively optimal solutions.

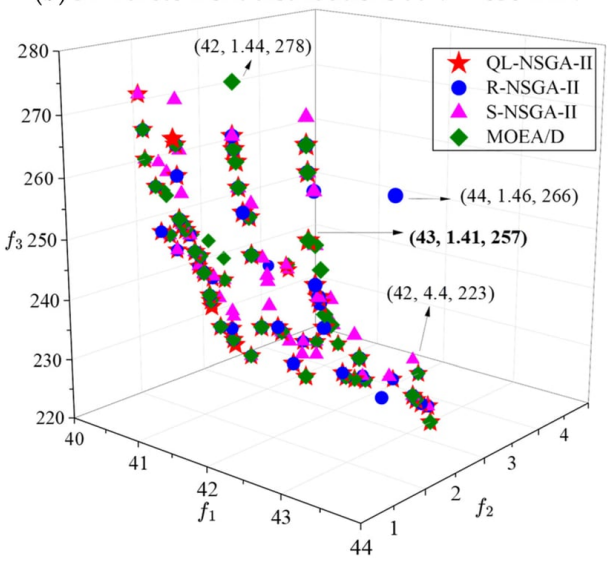
Combining the quantitative performance of the algorithms across different dimensions, it can be seen that QL-NSGA-II, relying on the adaptive operator scheduling of Q-Learning during evolution, explores relatively better solution sets than the comparison algorithms under each mode. This indicates that the proposed algorithm provides broader solution set coverage and stronger local exploration in large-scale problems, and can offer higher-quality decision schemes and richer trade-offs for actual PC component production.



(**a**) 3D Pareto front distributions at δ = 450 mm.



(**b**) 3D Pareto front distributions at δ = 650 mm.

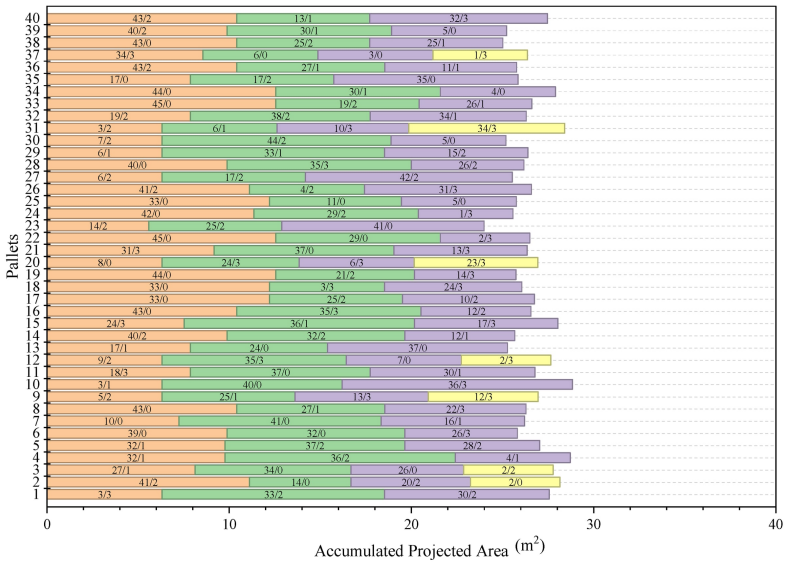


(**c**) 3D Pareto front distributions at δ = 800 mm.

Figure 8. 3D Pareto front distributions for P127 at different working ranges.

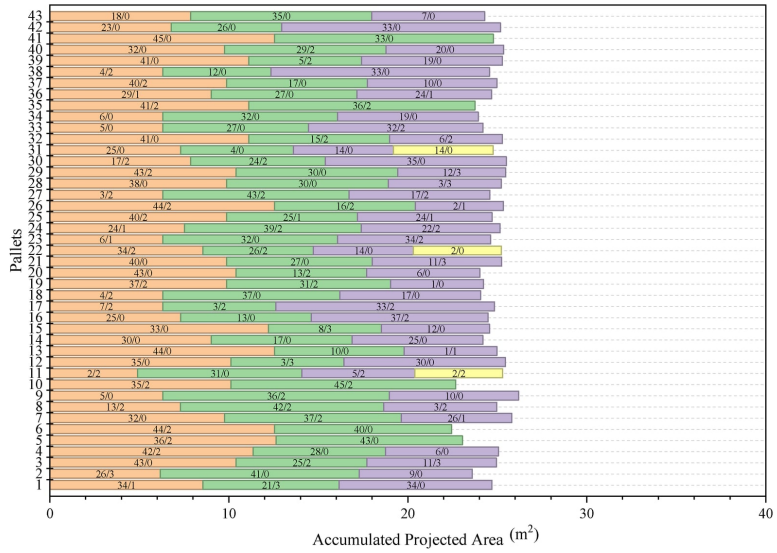
Figure 9a–c respectively show the pallet layout schemes for P127 under the $\delta = 650$ mm mode corresponding to the three minimum objective values. In the figure, the horizontal axis represents the effective area of a single pallet, which is 40 m^2, and the vertical axis represents the pallet index. Each colored segment represents one component, and the label format is "component ID/rotation angle," where 0, 1, 2, and 3 correspond to 0°, 90°, 180°, and 270°, respectively.

The scheme shown in Figure 9a uses only 40 pallets. The core feature of this scheme is that individual pallets are generally highly loaded, with most carrying 3 to 4 components and cumulative areas near 25–30 m^2. However, differences in cumulative occupied area among pallets are clear. Some pallets approach 30 m^2 in occupied area, while others are only about 18–20 m^2, intuitively reflecting the scheme's relatively high unbalancing load among pallets, and this strategy also does not reduce the number of climbing operations effectively. The scheme shown in Figure 9b uses 43 pallets. In sharp contrast to the scheme in Figure 9a, the occupied areas of the pallets in this scheme are notably consistent, with the cumulative occupied area of most pallets concentrated in the range of about 22–26 m^2 and a relatively narrow fluctuation range. Compared with the scheme in Figure 9a, this scheme trades an increase of 3 pallets for relatively stable takt times across workstations. At the same time, because the density of PC components on individual pallets is somewhat lower, component placement has more alternative space, and the number of climbing operations is also naturally alleviated to a certain extent. As shown in Figure 9c, this scheme also uses 43 pallets, but its layout configuration exhibits an organizational logic distinctly different from that of the scheme in Figure 9b. From the rotation angle labels of the components, it can be observed that the number of components with 0° rotation is notably higher in this scheme, indicating that when selecting component orientations, the algorithm tends to rotate regions with dense embedded parts toward the comfortable working zones near the two long sides of the pallet. Meanwhile, compared with the scheme in Figure 9b, the differences in cumulative occupied area among pallets increase to some extent. For example, the occupied area of an individual pallet, such as the first pallet, exceeds 30 m^2, while the area utilization of some other pallets is relatively low. This indicates that the selection of component positions and orientations is subject to human factors constraints, and, to some extent, the pursuit of layout compactness and load balance is abandoned in favor of effectively reducing the number of climbing operations.

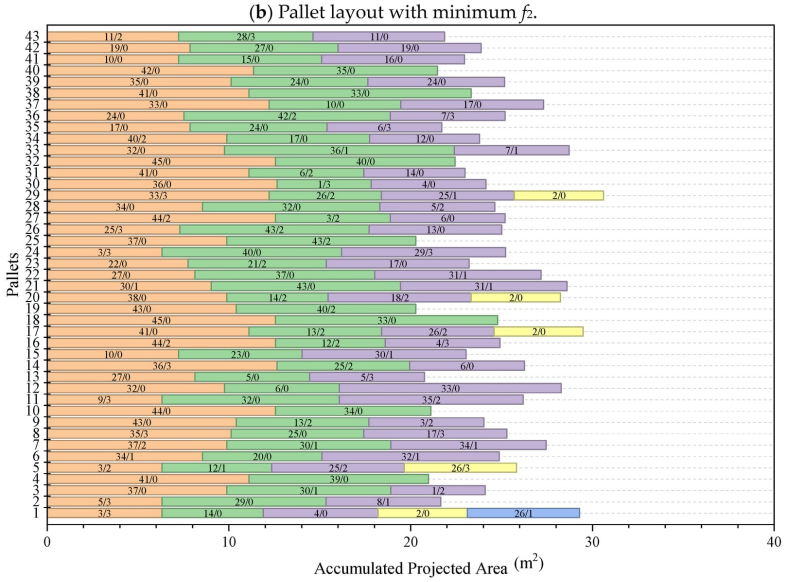


(**a**) Pallet layout with minimum f_1.

Figure 9. *Cont.*



(**b**) Pallet layout with minimum f_2.



(**c**) Pallet layout with minimum f_3.

Figure 9. Pallet layout schemes with minimum objectives for P127 at $\delta = 650$ mm.

The distinct characteristics of the three schemes indicate that the proposed algorithm can generate interpretable, diverse pallet layout schemes under different optimization priorities, providing a reference basis for flexible decision-making in factories across production scenarios.

6. Conclusions

To address the challenges of coordinating space utilization, production line balance, and worker safety in precast concrete flow-line production, this study developed a pallet layout model integrating human factors ergonomics and proposed a multi-objective evolutionary algorithm with Q-Learning adaptive guidance. The main conclusions are summarized in the following three points:

(1) A tri-objective mixed-integer programming model integrating human factors ergonomics was formulated for precast concrete pallet layout. By introducing the accessibility criteria of embedded parts on the pallet based on the effective working range, the risk of workers climbing onto pallets for process operations was quantified as an optimization objective. In the large-scale case, the diverse layout schemes generated

by the model effectively revealed the quantitative trade-off relationships among reducing pallet consumption, maintaining production line takt balance, and decreasing climbing operations, thereby providing a flexible decision-making space for various production priorities.

(2) The proposed QL-NSGA-II demonstrated favorable convergence and solution distribution uniformity in high-dimensional search spaces. In large-scale validations, the algorithm improved the hypervolume metric by 8.1% to 14.2% compared with benchmark algorithms. Meanwhile, the inverted generational distance and spacing metrics improved by 25.3% to 41.3% and 34.7%, respectively, and the number of non-dominated solutions increased by 28.37% to 49.16%. Rank-sum test p-values below 0.001 and effect sizes above 2.0 indicate that the online parameter scheduling mechanism is stable and effective in enhancing solution quality.

(3) The strategy selection of the reinforcement learning agent exhibited a dynamic evolution pattern characterized by early-stage global exploration, mid-stage bottleneck breakthrough, and late-stage local exploitation. Statistical tracking showed that the algorithm frequently invoked strong perturbation strategies early on to broaden the search scope, switched to objective-oriented mutation strategies to address specific bottlenecks during the middle stage, and increased the proportion of exploitation strategies later to consolidate high-quality solution sets. For complex layout tasks, the computation time increased by approximately 10% relative to the benchmark algorithm without adaptive mechanisms, which balances search accuracy and efficiency within an acceptable range of computation time.

Although this study has achieved favorable results in optimizing the pallet layout of precast components, certain limitations remain to be addressed in future work. First, the current model specifies that the components to be arranged are standard rectangular panels and primarily supports rotations in four orthogonal directions. Future research could introduce complex precast components, such as irregular polygonal geometric components. Second, the empirical validation was constrained to subsets of a single, albeit highly heterogeneous, production order. Future research will focus on testing the proposed framework across multiple diverse production orders and varying workshop layouts to fully validate its cross-scenario generalization capabilities. Furthermore, exploring end-to-end collaborative optimization frameworks utilizing advanced algorithms such as graph neural networks and deep reinforcement learning could help address dynamic production disturbances, including urgent order insertions and workstation processing time fluctuations, thereby enhancing the adaptability and overall operational efficiency of the production system.

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

Funding: This research was supported by the Luzhou Key Laboratory of Intelligent Construction and Low-carbon Technology, the Key Laboratory of Higher Education of Sichuan Province for Enterprise Informationalization and Internet of Things (grant number 2024WYY03) and the National Natural Science Foundation of China (Grant No. 51705386).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: Thanks to all the reference authors who gave us inspiration and help. The authors thank the editors and anonymous commentators for their valuable comments, which have improved the quality of this paper, and the experimental institutions that provide financial support for the paper.

Conflicts of Interest: Author Hansong Jiang was employed by the company Financial Investment Evaluation Center of Zigong High-tech Zone. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

Appendix A. Notations

Sets:

$N = \{1,2,...,n\}$: Set of PC types, indices are i or m.

$P_i = \{1,2,...,q_i\}$: Set of identical components of type i, indices are p or r.

(i,p): Unique identifier representing the p-th component of PC type i.

$K_i = \{1,2,...,k_i\}$: Set of embedded parts on PC component type i, index is k.

$J = \{1,2,...,B\}$: Set of candidate pallets on the assembly line. B is the upper bound.

Parameters:

L,W: Effective working length and width of the pallet.

l_i,w_i: Nominal length and width of component type i.

A_i: Single area of PC type i.

A_{total}: Total theoretical area of all components.

px_{ik},py_{ik}: Local design coordinates of embedded part relative to the component origin (at 0°).

d: Minimum clearance distance between components.

δ: Minimum clearance distance between components and pallet boundaries.

M: Big-M constant.

Decision Variables:

u_j: Whether pallet j is activated; 1 yes, 0 no.

$z_{ip,j}$: Whether component (i,p) is assigned to pallet j; 1 yes, 0 no.

$\theta_{ip}^{0}, \theta_{ip}^{90}, \theta_{ip}^{180}, \theta_{ip}^{270}$: Whether the component is rotated by 0°, 90°, 180°, or 270°; 1 yes, 0 no.

$\hat{l}_{ip}, \hat{w}_{ip}$: The actual bounding length and width after rotation.

x_{ip},y_{ip}: The coordinates of the lower left corner of the component after rotation.

S_j: Total component bottom area loaded on pallet j.

$\phi_{ip,mr}^{L/R/B/T}$: Indicates if a component is to the Left/Right/Bottom/Top of another; 1 yes, 0 no.

$gx_{ip,k},gy_{ip,k}$: The coordinates of the embedded part on the pallet.

$\xi_{ip,k}^{B/T}$: Whether the embedded part exceeds the reachable zone of the Bottom/Top aisles; 1 yes, 0 no.

$C_{ip,k}$: Whether the embedded part triggers an on-pallet climbing operation; 1 yes, 0 no.

References

1. Wang, Q.; Guo, W.; Xu, X.; Deng, R.; Ding, X.; Chen, T. Analysis of Carbon Emission Reduction Paths for the Production of Prefabricated Building Components Based on Evolutionary Game Theory. *Buildings* **2023**, *13*, 1557. [CrossRef]
2. Kim, H.; Kim, J.; Roh, S. The Embodied Life Cycle Global Warming Potential of Off-Site Prefabricated Concrete Products: Precast Concrete and Concrete Pile Production in Korea. *Buildings* **2023**, *13*, 2590. [CrossRef]
3. Xiong, F.; Chen, X.; Liu, H.; Xiong, M.; Wu, M. The Flow Shop Batch Scheduling Problem in a Prefabricated Component Manufacturing System with Two-Dimensional Bin Packing Constraints. *Int. J. Prod. Res.* **2025**, *63*, 7196–7237. [CrossRef]
4. Liang, J.; Liao, Y.; Liu, W.; Wu, L.; Wang, M. Research on the Talent Training Model for Master's Degree in Civil and Hydraulic Engineering under the Background of Intelligent Construction. *Int. J. Educ. Soc. Dev.* **2026**, *6*, 38–43. [CrossRef]
5. Wang, Z.; Hu, H. Improved Precast Production–Scheduling Model Considering the Whole Supply Chain. *J. Comput. Civ. Eng.* **2017**, *31*, 04017013. [CrossRef]
6. Liang, J.; Cao, Z.; Zu, Q.; Huang, H.; Guo, S. Production Sequencing and Layout Optimization of Precast Concrete Components under Mold Resource Constraints. *Buildings* **2024**, *14*, 3173. [CrossRef]

7. Liu, F.; Deng, X.; Peng, W.; Li, J.; Li, H.; Li, C. Scheduling Optimization for the Production of Prefabricated Components Considering Batch Production on Mold Tables. *Eng. Optim.* **2025**, *57*, 2960–2983. [CrossRef]
8. Warszawski, A. Production Planning in Prefabrication Plant. *Build. Environ.* **1984**, *19*, 139–147. [CrossRef]
9. Dawood, N.N.; Neale, R.H. A Capacity Planning Model for Precast Concrete Building Products. *Build. Environ.* **1993**, *28*, 81–95. [CrossRef]
10. Benjaoran, V.; Dawood, N. Intelligence Approach to Production Planning System for Bespoke Precast Concrete Products. *Autom. Constr.* **2006**, *15*, 737–745. [CrossRef]
11. Ray, B.; Ripley, P.; Neal, D. Lean Manufacturing—A Systematic Approach to Improving Productivity in the Precast Concrete Industry. *PCI J.* **2006**, *51*, 62–71. [CrossRef]
12. Li, X.; Li, Z.; Wu, G. Lean Precast Production System Based on the CONWIP Method. *KSCE J. Civ. Eng.* **2018**, *22*, 2167–2177. [CrossRef]
13. Wang, S.; Tang, J.; Zou, Y.; Zhou, Q. Research on Production Process Optimization of Precast Concrete Component Factory Based on Value Stream Mapping. *Eng. Constr. Archit. Manag.* **2020**, *27*, 850–871. [CrossRef]
14. Yuan, Z.; Qiao, Y.; Guo, Y.; Wang, Y.; Chen, C.; Wang, W. Research on Lean Planning and Optimization for Precast Component Production Based on Discrete Event Simulation. *Adv. Civ. Eng.* **2020**, *2020*, 8814914. [CrossRef]
15. Yin, S.Y.L.; Tserng, H.P.; Wang, J.C.; Tsai, S.C. Developing a Precast Production Management System Using RFID Technology. *Autom. Constr.* **2009**, *18*, 677–691. [CrossRef]
16. Altaf, M.S.; Bouferguene, A.; Liu, H.; Al-Hussein, M.; Yu, H. Integrated Production Planning and Control System for a Panelized Home Prefabrication Facility Using Simulation and RFID. *Autom. Constr.* **2018**, *85*, 369–383. [CrossRef]
17. Kim, M.K.; Wang, Q.; Park, J.W.; Cheng, J.C.P.; Sohn, H.; Chang, C.C. Automated Dimensional Quality Assurance of Full-Scale Precast Concrete Elements Using Laser Scanning and BIM. *Autom. Constr.* **2016**, *72*, 102–114. [CrossRef]
18. Tai, H.-W.; Chen, J.-H.; Cheng, J.-Y.; Hsu, S.-C.; Wei, H.-H. Learn Curve for Precast Component Productivity in Construction. *Int. J. Civ. Eng.* **2021**, *19*, 1179–1194. [CrossRef]
19. Liu, Y.; Tong, W.; Li, Q.; Yao, F.; Li, Y.; Li, H.X.; Huang, J. Study on Complexity of Precast Concrete Components and Its Influence on Production Efficiency. *Adv. Civ. Eng.* **2022**, *2022*, 9926547. [CrossRef]
20. Arashpour, M.; Kamat, V.; Bai, Y.; Wakefield, R.; Abbasi, B. Optimization Modeling of Multi-Skilled Resources in Prefabrication: Theorizing Cost Analysis of Process Integration in off-Site Construction. *Autom. Constr.* **2018**, *95*, 1–9. [CrossRef]
21. Chan, W.T.; Hu, H. Precast Production Scheduling with Genetic Algorithms. *Proc. IEEE Conf. Evol. Comput. ICEC* **2000**, *2*, 1087–1094. [CrossRef]
22. Chan, W.T.; Hu, H. An Application of Genetic Algorithms to Precast Production Scheduling. *Comput. Struct.* **2001**, *79*, 1605–1616. [CrossRef]
23. Chan, W.T.; Hu, H. Production Scheduling for Precast Plants Using a Flow Shop Sequencing Model. *J. Comput. Civ. Eng.* **2002**, *16*, 165–174. [CrossRef]
24. Leu, S.-S.; Hwang, S.-T. A GA-Based Model for Maximizing Precast Plant Production under Resource Constraints. *Eng. Optim.* **2001**, *33*, 619–642. [CrossRef]
25. Leu, S.-S.; Hwang, S.-T. GA-Based Resource-Constrained Flow-Shop Scheduling Model for Mixed Precast Production. *Autom. Constr.* **2002**, *11*, 439–452. [CrossRef]
26. Benjaoran, V.; Dawood, N.; Hobbs, B. Flowshop Scheduling Model for Bespoke Precast Concrete Production Planning. *Constr. Manag. Econ.* **2005**, *23*, 93–105. [CrossRef]
27. Chan, W.T.; Hu, H. Constraint Programming Approach to Precast Production Scheduling. *J. Constr. Eng. Manag.* **2002**, *128*, 513–521. [CrossRef]
28. Ko, C.-H.; Wang, S.-F. Precast Production Scheduling Using Multi-Objective Genetic Algorithms. *Expert Syst. Appl.* **2011**, *38*, 8293–8302. [CrossRef]
29. Tharmmaphornphilas, W.; Sareinpithak, N. Formula Selection and Scheduling for Precast Concrete Production. *Int. J. Prod. Res.* **2013**, *51*, 5195–5209. [CrossRef]
30. Yang, Z.; Ma, Z.; Wu, S. Optimized Flowshop Scheduling of Multiple Production Lines for Precast Production. *Autom. Constr.* **2016**, *72*, 321–329. [CrossRef]
31. Liu, Z.; Liu, Z.; Liu, M.; Wang, J. Optimization of Flow Shop Scheduling in Precast Concrete Component Production via Mixed-Integer Linear Programming. *Adv. Civ. Eng.* **2021**, *2021*, 6637248. [CrossRef]
32. Xiong, F.; Chu, M.; Li, Z.; Du, Y.; Wang, L. Just-in-Time Scheduling for a Distributed Concrete Precast Flow Shop System. *Comput. Oper. Res.* **2021**, *129*, 105204. [CrossRef]
33. Ma, Z.; Yang, Z.; Liu, S.; Wu, S. Optimized Rescheduling of Multiple Production Lines for Flowshop Production of Reinforced Precast Concrete Components. *Autom. Constr.* **2018**, *95*, 86–97. [CrossRef]
34. Wang, Z.; Hu, H.; Gong, J. Framework for Modeling Operational Uncertainty to Optimize Offsite Production Scheduling of Precast Components. *Autom. Constr.* **2018**, *86*, 69–80. [CrossRef]

35. Du, J.; Dong, P.; Sugumaran, V. Dynamic Production Scheduling for Prefabricated Components Considering the Demand Fluctuation. *Intell. Autom. Soft Comput.* **2020**, *26*, 715–723. [CrossRef]
36. Wang, Z.; Liu, Y.; Hu, H.; Dai, L. Hybrid Rescheduling Optimization Model under Disruptions in Precast Production Considering Real-World Environment. *J. Constr. Eng. Manag.* **2021**, *147*, 04021012. [CrossRef]
37. Kim, T.; Kim, Y.W. Proactive Production Scheduling Approach for Off-Site Construction with Due Date Uncertainty. *Appl. Sci.* **2024**, *14*, 11017. [CrossRef]
38. Yang, R.; Xu, S.; Li, H.; Zhu, H.; Zhao, H.; Wang, X. Action-Oriented Deep Reinforcement Learning Method for Precast Concrete Component Production Scheduling. *Buildings* **2025**, *15*, 697. [CrossRef]
39. Man, T.; Osipov, V.Y.; Zhukova, N.; Subbotin, A.; Ignatov, D.I. Neural Networks for Intelligent Multilevel Control of Artificial and Natural Objects Based on Data Fusion: A Survey. *Inf. Fusion* **2024**, *110*, 102427. [CrossRef]
40. Prata, B.d.A.; Pitombeira-Neto, A.R.; Sales, C.J.d.M. An Integer Linear Programming Model for the Multiperiod Production Planning of Precast Concrete Beams. *J. Constr. Eng. Manag.* **2015**, *14*, 04015029. [CrossRef]
41. Hu, H. A Study of Resource Planning for Precast Production. *Archit. Sci. Rev.* **2007**, *50*, 106–114. [CrossRef]
42. Khalili, A.; Chua, D.K. Integrated Prefabrication Configuration and Component Grouping for Resource Optimization of Precast Production. *J. Constr. Eng. Manag.* **2014**, *140*, 04013052. [CrossRef]
43. Wang, D.; Liu, G.; Li, K.; Wang, T.; Shrestha, A.; Martek, I.; Tao, X. Layout Optimization Model for the Production Planning of Precast Concrete Building Components. *Sustainability* **2018**, *10*, 1807. [CrossRef]
44. Han, Z.; Yu, L.; Li, B.; Qi, Y. Research on Prefabricated Component Production Line Mold Platform Combination Method. *J. Phys. Conf. Ser.* **2022**, *2187*, 012059. [CrossRef]
45. Xu, C.; Zhou, X.; Lin, X.C.; Peng, W.; Zhang, C. Automated Optimization for Steel Mold of Precast Components Based on Multi-Objective Evolutionary Algorithm. *J. Constr. Steel Res.* **2024**, *221*, 108914. [CrossRef]
46. Zou, Y.; Gao, Q.; Wang, S. Optimization of the Stacking Plans for Precast Concrete Slab Based on Assembly Sequence. *Buildings* **2022**, *12*, 1538. [CrossRef]
47. Marasini, R.; Dawood, N.S.; Hobbs, B. Stockyard Layout Planning in Precast Concrete Products Industry: A Case Study and Proposed Framework. *Constr. Manag. Econ.* **2001**, *19*, 365–377. [CrossRef]
48. Tan, Y.; Yu, D.; Yang, S.; Gai, T.; Wang, C. Precast Concrete Component Yard Layout Optimization Considering Safety and Efficiency. *J. Asian Archit. Build. Eng.* **2025**, *24*, 2800–2816. [CrossRef]
49. Fırat, H.; Alpaslan, N. An Effective Approach to the Two-Dimensional Rectangular Packing Problem in the Manufacturing Industry. *Comput. Ind. Eng.* **2020**, *148*, 106687. [CrossRef]
50. Kim, T.; Kim, Y.W.; Lee, D.; Kim, M. Reinforcement Learning Approach to Scheduling of Precast Concrete Production. *J. Clean. Prod.* **2022**, *336*, 130419. [CrossRef]
51. Sengupta, A.K.; Das, B. Maximum Reach Envelope for the Seated and Standing Male and Female for Industrial Workstation Design. *Ergonomics* **2000**, *43*, 1390–1404. [CrossRef] [PubMed]
52. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; Routledge: New York, NY, USA, 2013; ISBN 9781134742707.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.