Networks and Heterogeneous Media

https://www.aimspress.com/journal/nhm

NHM, 21(4): 1172–1196.
DOI: 10.3934/nhm.2026047
Received: 18 March 2026
Revised: 05 May 2026
Accepted: 05 June 2026
Published: 10 June 2026

Research article

A hierarchical decoupling framework and hybrid adaptive genetic algorithm for multimachine dual-arm PCB placement scheduling

Chang Kou[1], **Mingze Sun**[2], **Ting Song**[2] **and Jihong Shen**[2,*]

[1] Dalian Rijia Co., Ltd., Dalian 116600, China
[2] College of Mathematical Sciences, Harbin Engineering University, Harbin 150000, China

* **Correspondence:** Email: shenjihong@hrbeu.edu.cn; Tel: 045182519754; Fax: 045182519754.

Abstract: Printed circuit boards (PCBs) are an indispensable part of electronic products. As the consumer electronics market continues to expand, the PCB assembly rate has become a key factor limiting production line efficiency. PCB placement scheduling is a highly complex, multivariable combinatorial optimization problem characterized by nonlinearity and high coupling. From a system perspective, this scheduling problem is modeled as a complex network optimization, wherein the intricate coupling of resources, time, and processes parallels the node interdependencies and information propagation mechanisms inherent in complex system architectures. Given its nondeterministic polynomial-time (NP)-hard nature and the dynamic constraints of the production environment, this study develops a rigorous mathematical model based on motion trajectory analysis and proposes a three-stage hierarchical decoupling framework. This decomposition transforms the high-dimensional network complexity into manageable subtasks while preserving essential variable interdependencies. To solve this complex network optimization, the framework integrates a hybrid adaptive genetic algorithm (HAGA) implemented through a three-level strategy.Initially, an improved adaptive genetic algorithm (IAGA) handles global load balancing; subsequently, an improved spectral clustering algorithm groups components to minimize tool-changing overhead; finally, a hybrid genetic algorithm and ant colony optimization (GA-ACO) optimizes synchronized intramachine paths. Unlike partial-stage methods, this holistic approach ensures synergistic optimization from global network equilibrium to local machine precision. Numerical experiments using real-world data from Dalian Rijia Co., Ltd. demonstrate that HAGA achieves at least a 10% reduction in the production cycle. This research proves that the hierarchical strategy effectively balances computational efficiency with global optimization performance.

Keywords: printed circuit board; multimachine dual-arm system; hybrid adaptive genetic algorithm (HAGA); spectral clustering algorithm; production line load balancing

1. Introduction

In the Industry 4.0 era, surface mount technology (SMT) serves as the backbone of electronics manufacturing. Within SMT production lines, the placement process remains the primary bottleneck, accounting for 65–70% of total processing time and over 60% of total capital investment. Consequently, optimizing placement machine utilization is essential for streamlining assembly and enhancing throughput.

Despite extensive research on printed circuit board (PCB) assembly scheduling, critical challenges remain in heterogeneous multimachine dual-arm systems, particularly regarding high coupling between constraints, workload imbalance, and inefficient path planning. Traditional metaheuristic algorithms often struggle with local optima and poor scalability due to the lack of systematic network modeling and hierarchical decomposition.

Fundamentally, the dual-arm PCB systems can be conceptualized as a placement scheduling network (PSN), in which placement points and feeder slots act as functional nodes, and motion trajectories serve as dynamically weighted edges. The coordinated scheduling of resources, time, and processes is consistent with node interactions and information propagation in complex networks. Nevertheless, traditional metaheuristics often fall into local optima when facing high coupling and nonlinear constraints such as nozzle compatibility and collision avoidance, as they cannot effectively mine the community structure inside the network.

To address these limitations, this research interprets the scheduling task as a complex network optimization problem and develops a hierarchical decoupling framework based on a hybrid adaptive genetic algorithm (HAGA). A core innovation is the formulation of a rigorous mathematical model through systematic motion trajectory analysis, capturing complex network dynamics to minimize makespan. The proposed architecture decomposes the high-dimensional network complexity into three interrelated stages:

1. **Global load balancing (network constraint propagation):** An improved spectral clustering algorithm (IAGA) is designed for component-to-machine allocation. By minimizing the theoretical cycle time difference through a load-balance-oriented fitness function, this stage determines the component subset for each machine, effectively defining the search space boundary for the subsequent stages.

2. **Intelligent component grouping (structural complexity reduction):** Acting as a bridge, an improved spectral clustering algorithm is constructed to build a similarity matrix considering spatial constraints and nozzle compatibility. This transforms the large-scale disordered problem into manageable subtasks, reducing the computational burden for path planning while minimizing tool-changing overhead.

3. **Intramachine path planning (dynamic execution):** As the execution layer, a hybrid genetic algorithm and ant colony optimization (GA-ACO) algorithm is employed for the synchronized coordination of arm allocation and path planning. This stage converts the logical allocation and grouping strategies into the actual minimum physical makespan.

2. Related work

PCB placement scheduling has long been a core combinatorial optimization problem in SMT assembly lines. Early research focused on sequential tasks for single-gantry machines, including component sequencing, feeder arrangement, and nozzle assignment [1, 2]. Later studies transitioned to multihead configurations, where researchers developed efficient pick-and-place sequences to improve operational efficiency [2]. As complexity grew, the interdependency between subproblems became a focal point, and integer programming or hybrid genetic algorithms were adopted to handle nozzle assignment and component sequencing [3, 4]. To further reduce placement time, various metaheuristic methods were introduced, such as improved particle swarm optimization, the best unified algorithm, and multiswarm firefly algorithms [5–7]. Furthermore, studies began incorporating physical constraints and advanced neural networks to achieve high-efficiency operations in complex placement scenarios [8–10]. However, these methods predominantly treated machines as isolated entities, failing to capture the cascading constraint propagation and network-level interference that occur across heterogeneous multimachine production lines.

The introduction of dual-gantry systems shifted the research emphasis from single-machine efficiency to workload balancing and motion synchronization. To reduce idle time between the front and rear arms, multiphase heuristics and adaptive clustering genetic algorithms were proposed [11, 12]. Concurrently, adaptive tabu search and greedy genetic algorithms were developed to optimize the entire work cycle, including component allocation and feeder arrangement [13, 14]. For more flexible production, modular placement configurations were also formulated via integer programming to handle multiproduct scenarios [15]. Although these approaches mitigated intramachine idle time, they predominantly relied on static heuristic rules or greedy strategies that could not dynamically decouple the intricate interdependencies between dual-arm collaboration and upstream component allocation, thereby struggling to maintain global efficiency when scaling up to heterogeneous multimachine environments.

At the production line level, the core challenge lay in global component distribution and line balancing to eliminate bottlenecks. Researchers adopted divide-and-conquer strategies, as studied in early foundational works [16, 17] and their later extensions [18–21].These strategies decompose the complicated PCB assembly scheduling problem into several independent subproblems including component allocation, feeder arrangement and pick-and-place sequencing. Each subproblem is solved separately with tailored heuristics or optimization algorithms to reduce computational complexity. However, most of them optimize sub-tasks in isolation and neglect the tight coupling among constraints in heterogeneous multi-machine systems. Immune algorithms and hybrid genetic algorithms were verified to be effective in shortening assembly time [22–24]. Through linear programming and hierarchical methods, multilevel scheduling frameworks were gradually established to improve overall efficiency [25–27]. Recently, research has moved toward large-scale collaborative optimization and multiobjective performance. Min-max models and grouping heuristics were designed for heterogeneous lines [28–30], and nonproduction factors such as energy consumption, equipment load balance, and maintenance cost have been actively integrated into optimization objectives [31–33]. Although these multiobjective extensions provided a broader systemic perspective, the underlying optimization engines typically handled global allocation, component grouping, and path planning as fragmented, independent subproblems. This decoupled approach failed to resolve the high-dimensional coupling among spatial-nozzle-machine heterogeneity, often resulting in theoretical local optima that could not translate into physically executable and collision-free motion trajectories.

In recent years, complex network theory has provided new perspectives for scheduling modeling by describing node relationships and constraint propagation mechanisms. In the context of intelligent manufacturing, deep reinforcement learning and divide-and-combine heuristics have been applied to multiproduct scheduling scenarios [34, 35]. However, most studies only regard the scheduling problem as a general combinatorial optimization issue without fully utilizing its topological structure and dynamic characteristics. Despite the above progress, most existing methods only focus on partial subproblems and lack a systematic hierarchical decoupling framework for heterogeneous multimachine dual-arm systems. The high coupling between nozzle constraints, spatial positions, and machine heterogeneity remains unsolved effectively. The heterogeneous and high-dimensional characteristics of dual-arm systems further make this problem more difficult to solve with conventional methods, meaning traditional algorithms struggle to balance computational efficiency and global optimization performance in large-scale scenarios.

3. Problem description and mathematical modeling

3.1. Problem description

The multimachine dual-arm PCB placement scheduling problem is formally defined as a constrained combinatorial optimization problem. Given a set of components, a series of heterogeneous dual-arm placement machines, nozzle compatibility constraints, spatial positions, and motion characteristics, the objective is to minimize the maximum completion time (makespan) of the entire production line. The problem involves three strongly coupled subproblems:

- Global load balancing: Distributing the components among the K machines such that the workload (expected placement time) is equilibrated, accounting for machine-specific kinematic constraints and component-nozzle compatibility.
- Component clustering: For each machine, grouping the assigned components into pick-and-place cycles. Each cycle must respect the maximum head capacity and nozzle constraints of the dual arms.
- Path optimization: Determining the optimal sequence of picking operations from the feeders and placement operations on the PCB to minimize the total nonproductive travel distance of the gantry arms.

In order to simplify the mathematical modeling process and align with the actual production requirements of the factory, we have conducted the following analysis of the specific situation of the factory.

- Deterministic processing times: The picking, placement, and nozzle replacement durations are constant for each specific machine and component type, as defined in the kinematic parameters.
- Constant arm velocity: The movement of the gantry arm between any two points (x_1, y_1) and (x_2, y_2) is modeled using an average operational speed, neglecting the transient effects of acceleration and deceleration.
- Component availability: All required components are preloaded into the designated feeder slots, and there is no shortage of materials during the assembly of a single PCB batch.
- Fixed placement origin: The PCB is accurately positioned by the mechanical stopper,

and the coordinate transformation (origin compensation) remains valid throughout the entire production process.

- Nozzle compatibility: Each placement head can only carry one nozzle at a time, and the nozzle must be compatible with the physical dimensions and vacuum requirements of the component being picked.
- Interference-free operation: For dual-arm machines, it is assumed that the control system manages collision avoidance, allowing the optimization to focus on the trajectory efficiency of each arm independently within the hierarchical framework.
- Negligible PCB transfer time and machine setup time: For simplification and consistency with actual factory conditions, PCB transfer time between machines and machine setup time are regarded as fixed constants and are NOT included in the makespan calculation.

3.2. *Definition of parameters and variables*

To facilitate the mathematical formulation of the PCB placement scheduling problem, the parameters and variables used in this paper are defined in Table 1.

Table 1. Definition of parameters, constants, and variables used in the optimization model.

Symbol	Description
Indices and sets	
k	Index of machines ($k = 1, 2, \ldots, K$).
g	Index of machine arms ($g \in \{F, R\}$, F: front, R: rear).
h	Index of placement heads ($h = 1, 2, \ldots, H_k$).
c, q	Index of pickup and placement cycles ($c, q = 1, 2, \ldots, C_k^g$).
t	Index of component types ($t = 1, 2, \ldots, T$).
m	Index of nozzle types ($m = 1, 2, \ldots, M_{\text{total}}$).
i, j	Indices for feeder pickup position and PCB placement position.
Ω	Set of all feasible allocation schemes.
Parameters and constants	
a	Time required to replace a single nozzle on the ANC (s).
v_f	Average moving speed of the manipulator (mm s^{-1}).
f_c	Speed coefficient of cycle c.
t_z	Time for z-direction movement during pickup and placement (s).
N	Total number of components.
N_k	Number of components assigned to machine k.
N_t	Number of components of type t.
λ_m	Matching degree between component type t and nozzle type m.
$NZ(t)$	Nozzle type required for component type t.
ω, ρ	Penalty coefficients for nozzle changes and pickup sequences.
H_k	Total number of placement heads on machine k.
Operational variables	
S_c^0, S_c^l	Positions of the first and last components placed in cycle c (mm).
A_c^0	Position of the first nozzle replacement on the ANC in cycle c (mm).
φ_k^g	Total number of cycles on arm g of machine k.
u	Number of components picked up by a nozzle on the placement head.
EP, EW	Total pickup and placement times for a single arm cycle (s).
$\mu_{k\theta}$	Number of nozzle replacements between two adjacent cycles.
$\mathcal{W}$	A specific component allocation scheme (chromosome).
Decision variables	
x_{ij}	Binary: 1 if component j is placed at position i; 0 otherwise.
$\gamma_{(c-1)c}$	Binary: 1 if a nozzle change occurs between cycles $c-1$ and c; 0 otherwise.
X_{chm}^{kg}	Binary: 1 if nozzle m is assigned to head h, arm g, machine k in cycle c.
Z_{cht}^{kg}	Binary: 1 if component t is assigned to head h, arm g, machine k in cycle c.

3.3. Mathematical modeling

The primary optimization objective is to minimize the makespan of the production line. From a complex network perspective, this is equivalent to optimizing the information propagation and task execution efficiency across the PSN. The PSN is characterized by its nonuniform topology, where the coupling between nozzle compatibility and spatial constraints creates a heterogeneous constraint propagation environment. The intricate coordination of resources, time, and processes mirrors the node interactions inherent in complex system architectures. The objective function is defined as the maximum working time among all machine nodes:

$$\min f = \max_{k \in \{1,2,\ldots,K\}} T_k(\theta), \quad \theta \in \Omega. \tag{3.1}$$

The total working time $T_k(\theta)$ for a specific machine node k represents the cumulative weight of all active edges (picking, placing, and moving) within its local subnetwork.

3.3.1. Operational constraints

The optimization is subject to the following constraints:

1) **Line load balancing:** To ensure high throughput, the workload difference between any two machines ω and ξ must remain within a threshold ϵ:

$$|T_\omega - T_\xi| < \epsilon, \quad \forall \omega, \xi \in \{1, 2, \ldots, K\}. \tag{3.2}$$

2) **Dual-arm synchronization:** The difference in the number of pick-and-place cycles between the front arm (C_F) and the rear arm (C_R) is restricted to maintain collaborative efficiency:

$$0 \le C_k^F - C_k^R \le 1. \tag{3.3}$$

3) **Minimum cycle requirement:** The number of cycles for each machine arm must be an integer and satisfy the physical lower bound:

$$C_k \ge \left\lceil \frac{N_k}{H_k} \right\rceil, \quad k = 1, 2, \ldots, K. \tag{3.4}$$

3.3.2. Assignment and logic constraints

4) **Component-nozzle logical coupling:** The following constraints utilize a Big-M approach to ensure that a component type t can only be assigned to a head h if the corresponding nozzle m is equipped:

$$\sum_{c=1}^{C_k^g} \sum_{t=1}^{T} Z_{cht}^{kg} \cdot X_{chm}^{kg} \le T \cdot \sum_{c=1}^{C_k^g} X_{chm}^{kg}, \tag{3.5}$$

$$\sum_{c=1}^{C_k^g} \sum_{t=1}^{T} Z_{cht}^{kg} \cdot X_{chm}^{kg} \ge \sum_{c=1}^{C_k^g} X_{chm}^{kg}. \tag{3.6}$$

5) **Component and nozzle coverage:** These constraints ensure that all components are assigned (Eq (3.7)) and each head is equipped with exactly one nozzle type per cycle (Eq (3.8)):

$$\sum_{g=F,R} \sum_{h=1}^{H_k^g} \sum_{c=1}^{C_k^g} Z_{cht}^{kg} = N_t, \quad \forall t, \tag{3.7}$$

$$\sum_{m=1}^{M_{\text{total}}} X_{chm}^{kg} = 1, \quad \forall k, g, c, h. \tag{3.8}$$

6) **Capacity and process integrity:** The number of nozzles per cycle (Eq (3.9)), the placements per cycle (Eq (3.10)), and the total PCB coverage (Eq (3.11)) must satisfy the equipment limits:

$$\sum_{m=1}^{M_{\text{total}}} \sum_{h=1}^{H_k^g} X_{chm}^{kg} \in [1, H_k^g], \quad \forall k, g, c, \tag{3.9}$$

$$0 \leq \phi_{kg}^c \leq H_{kg}, \quad \forall k, g, c, \tag{3.10}$$

$$\sum_{m,g,k,h,c} X_{chm}^{kg} = N. \tag{3.11}$$

7) **Placement specificity:** Each component type uses at most one nozzle type per head, and physical matching between the component and nozzle must be maintained:

$$\sum_{h=1}^{H_k^g} X_{chm}^{kg} \leq 1, \quad \forall k, g, c, m, \tag{3.12}$$

$$\sum_{t=1}^{T} Z_{cht}^{kg} \cdot NZ(t) = \sum_{m=1}^{M_{\text{total}}} X_{chm}^{kg} \cdot m, \quad \forall k, g, c, h. \tag{3.13}$$

3.3.3. Path optimization constraints

8) **Placement path uniqueness:** To define a valid sequence (traveling salesman problem (TSP) structure), each placement point must have exactly one preceding and one succeeding point:

$$\sum_{i=1}^{H} x_{ij} = 1, \quad j = 1, 2, \ldots, H, \tag{3.14}$$

$$\sum_{j=1}^{H} x_{ij} = 1, \quad i = 1, 2, \ldots, H-1. \tag{3.15}$$

3.3.4. Topological characteristics of the PSN

The efficiency of PCB assembly is fundamentally governed by the topological properties of the PSN. Analyzing these characteristics provides a theoretical basis for the proposed hierarchical decoupling framework:

1. **Node degree and connectivity:** In the PSN, nodes (placement points and feeder slots) are interconnected by potential motion trajectories. The **node degree distribution** is highly non-uniform due to the heterogeneity of machine configurations. Nodes requiring specific nozzle types exhibit lower functional connectivity, effectively creating "bottleneck" regions within the network topology.

2. **Clustering coefficient:** The PSN displays a high **average clustering coefficient**. This is driven by the spatial proximity of components on the PCB; placement points located in the same coordinate quadrant tend to form localized "cliques" of tasks that can be executed with minimal gantry arm movement.

3. **Modularity and community structure:** The most prominent feature of the PSN is its strong **modularity**. The intricate coupling of nozzle compatibility and spatial constraints naturally partitions the network into distinct communities. Nodes within a community share high edge weights (representing low travel time and zero tool-changing overhead), and intercommunity edges are "weak" due to the necessity of nozzle replacement.

The proposed improved spectral clustering algorithm is mathematically congruent with these topological features. By performing eigendecomposition on the normalized Laplacian matrix, the algorithm effectively identifies the community structures within the PSN. This approach transforms the high-dimensional, coupled scheduling task into a series of dense subnetwork optimizations, ensuring that the resulting pick-and-place cycles maximize intracommunity execution parallelism while minimizing intercommunity transitions.

4. Solution to the optimal PCB scheduling model based on hybrid adaptive genetic algorithm

4.1. Proposed hierarchical decoupling framework

Multimachine dual-arm PCB assembly scheduling is a high-dimensional nondeterministic polynomial-time (NP)-hard problem, featuring intricate coupling of component allocation, nozzle selection, and placement sequencing. To address these challenges, this study first constructs a **complete mathematical model** based on kinematic trajectory analysis, aiming to minimize global makespan. However, direct solution of this comprehensive model is computationally prohibitive due to the vast search space and operational interdependencies of dual-arm systems. Consequently, a **hierarchical decoupling framework** is proposed to decompose the global optimization problem into three interrelated subproblems.

This framework solves the mathematical model via the following workflow: First, an **IAGA** distributes component types across machines to satisfy load balancing constraints, establishing a balanced workload foundation. Building on these results, an **improved spectral clustering algorithm** aggregates placement points by spatial proximity and nozzle constraints to minimize tool-changing overhead. Finally, a **hybrid GA-ACO strategy** optimizes precise motion trajectories: An improved genetic algorithm determines arm task assignment and cycle sequencing, and an improved ant colony algorithm fine-tunes intracycle paths. Feeder positions are concurrently determined based on the minimum distance principle to reduce nonproductive travel. The integrated algorithmic workflow of the proposed HAGA is visually illustrated in Figure 1, which clearly presents the three-stage hierarchical optimization logic based on similar data of a single machine.

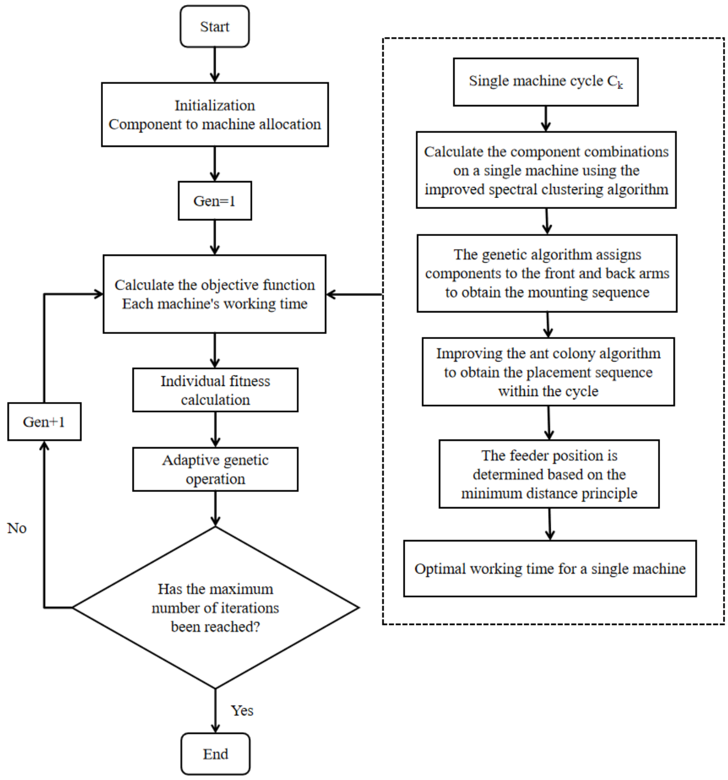


Figure 1. Three-stage hierarchical decoupling workflow of the proposed hybrid adaptive genetic algorithm (HAGA) for multimachine dual-arm PCB placement scheduling.

4.2. Component type allocation via IAGA

The allocation of component types to specific machines constitutes a complex combinatorial optimization problem. Traditional GAs often suffer from premature convergence due to fixed operator probabilities and simple reciprocal fitness functions that fail to distinguish between individuals with similar objective values. To address these limitations, we propose an IAGA. This algorithm integrates a constraint-aware encoding scheme with a dynamic fitness scaling mechanism and, crucially, employs an adaptive strategy that adjusts crossover (P_c) and mutation (P_m) probabilities based on population diversity.

4.2.1. Problem representation and objective evaluation

To effectively map the physical allocation problem into the algorithmic domain, we first consider the hardware layout. As shown in Figure 2, the topological configuration of the tandem SMT production line consists of K heterogeneous placement machines connected in series. Each machine has specific nozzle configurations and feeder capacities, which impose strict compatibility constraints on component allocation.

Based on this physical structure, we utilize an integer encoding scheme, shown in Figure 3. The chromosome is represented as a vector where each gene corresponds to a component type, and the gene value represents the assigned machine ID.

To evaluate these chromosomes, we transform the minimization of the production cycle time (f) into a maximization problem. A linear scaling fitness function is introduced to amplify the differences between high-performing and average individuals, thereby enhancing evolutionary pressure:

$$F(i) = \frac{f_{\max} - f_i}{f_{\max} - f_{\text{avg}}}, \tag{4.1}$$

where $F(i)$ is the fitness of the ith chromosome, f_i is its corresponding production cycle time, and $f_{\max}$ and f_{avg} represent the maximum and average cycle times within the current population, respectively.

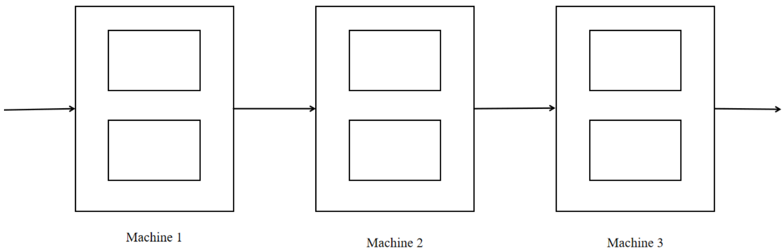


Figure 2. Topological configuration of the tandem SMT production line system.

Machine	1	2	1	3	2	1	1	3	2	1
Component Type	1	2	3	4	5	6	7	8	9	10

Figure 3. Integer-based chromosome encoding scheme for component-to-machine allocation.

4.2.2. Adaptive evolutionary mechanism

The adaptive crossover and mutation probabilities are dynamically adjusted based on population fitness and diversity. This mechanism balances global exploration and local exploitation, effectively avoiding premature convergence. The adaptive formulas are defined as follows:

$$M = \max\left(0, \min\left(1, 1 - \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(F_i - F_{\text{avg}})^2}}{F_{\max} - F_{\min}}\right)\right), \tag{4.2}$$

$$P_c = \begin{cases} \left(1 - \frac{F' - F_{\text{avg}}}{F_{\max} - F_{\text{avg}}}\right) \times M, & F' \geq F_{\text{avg}}, \\ \left(0.5 + \frac{F' + F_{\text{avg}}}{2(F_{\text{avg}} - F_{\min})}\right) \times M, & F' < F_{\text{avg}}, \end{cases} \tag{4.3}$$

where F' denotes the fitness value of the superior parent individual in the crossover operation.

$$P_m = \begin{cases} \left(1 - \frac{F - F_{\text{avg}}}{F_{\max} - F_{\text{avg}}}\right) \times M, & F \geq F_{\text{avg}}, \\ \left(0.5 + \frac{F + F_{\text{avg}}}{2(F_{\text{avg}} - F_{\min})}\right) \times M, & F < F_{\text{avg}}. \end{cases} \tag{4.4}$$

4.2.3. Heuristic constraint handling

Two-step repair ensures offspring feasibility: 1) Feeder slot constraint repair: reallocate over-capacity component types to machines with available slots; 2) Idle machine repair: reassign compatible components to idle machines for load balancing.

The convergence of the IAGA is ensured by its adaptive evolutionary mechanism. By dynamically adjusting the crossover (P_c) and mutation (P_m) probabilities based on the population diversity factor M, the algorithm effectively maintains a balance between exploration and exploitation. Experimental results indicate that the population fitness exhibits a monotonic upward trend and stabilizes after a finite number of iterations, proving the algorithm's robust convergence in global load balancing.

4.3. Component grouping and cycle allocation based on improved spectral clustering

This section addresses the subproblem of pick-and-place (P&P) cycle allocation using an enhanced spectral clustering approach. Traditional spectral clustering primarily relies on Euclidean distance to measure similarity and performs dimensionality reduction through eigendecomposition. However, in the context of PCB assembly, low-dimensional embeddings often lack intuitive physical meaning, making it difficult to directly incorporate heterogeneous constraints such as nozzle compatibility, which can lead to fragmented clustering results. We propose an optimized algorithm that integrates spatial proximity, component type similarity, and nozzle compatibility into the adjacency matrix, grouping suitable components to maximize simultaneous pickup and reduce arm movements and tool-changing overhead for high-throughput production. The steps are as follows:

- **Multidimensional weight matrix construction:** Integrates spatial proximity, component type similarity, and nozzle compatibility into a similarity matrix W, capturing the intricate node interdependencies of the PSN.
- **Recursive spectral clustering with capacity control:** Employs eigen-decomposition and recursive partitioning to achieve dynamic cluster capacity control, ensuring each group respects the physical constraints of the placement heads.
- **Heuristic nearest-neighbor merging:** Resolves unassigned nodes via a nearest-neighbor search to valid cluster centers, maintaining global assignment integrity while preserving spatial compactness.

The convergence of the improved spectral clustering algorithm is rooted in the properties of the Laplacian matrix L_{sym}. Because L_{sym} is a symmetric positive semidefinite matrix, its eigendecomposition yields a stable and unique low-dimensional embedding of the PSN. The recursive partitioning process and the deterministic nearest-neighbor merging guarantee that the algorithm converges to a stable community structure within polynomial time.

4.4. Internal collaborative work assignment via hybrid GA-ACO

The hybrid GA-ACO optimizes arm allocation and path planning: The GA optimizes intercycle sequence, and ACO fine-tunes the intracycle path, with feeder positions determined by the minimum-distance greedy principle.

4.4.1. Chromosome representation and decoding

As shown in Figure 4, double-chain real-value encoding includes the placement sequence chain and the component-type chain, with cycles split between front/rear arms ($\lceil C_k/2 \rceil / \lfloor C_k/2 \rfloor$). As illustrated in Figure 5, nozzle assignment encoding sets idle head gene values to zero.

	front arm									rear arm					
	Loop 1			Loop 2			Loop 3			Loop 1			Loop 2		
Placement Location	1	12	14	2	11	13	3	6	0	7	8	4	5	9	10
Component Type	1	1	1	2	4	4	3	3	0	1	2	1	2	1	2

Figure 4. Chromosomal encoding for gantry arm cluster allocation.

Nozzle type	1	1	1	1	4	4	3	3	0	2	2	2	1	1	1

Figure 5. Nozzle assignment encoding scheme for multihead placement modules.

4.4.2. Initialization and hybrid path optimization strategy

Hybrid leader-follower architecture: The GA optimizes intercycle sequence (minimizing nozzle changes, Figure 6), and ACO solves intracycle TSP with 2-opt local search (Figure 7). Key equations:

- Ant movement probability:

$$P_{ij}^k(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^\alpha \left[\eta_{ij}(t)\right]^\beta}{\sum_{s \in A_k} \left[\tau_{is}(t)\right]^\alpha \left[\eta_{is}(t)\right]^\beta}, & j \in A_k, \\ 0, & \text{otherwise}. \end{cases} \tag{4.5}$$

- Pheromone update:

$$\tau_{ij}(g+1) = (1-\rho)\tau_{ij}(g) + \sum_{k=1}^{m} \Delta\tau_{ij}^k. \tag{4.6}$$

- 2-opt local search condition:

$$d(i, j) + d(i+1, j+1) < d(i, i+1) + d(j, j+1). \tag{4.7}$$

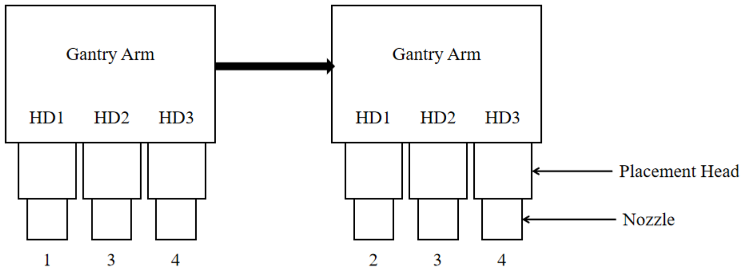


Figure 6. Schematic representation of the automated nozzle change process on the automatic nozzle changer (ANC).

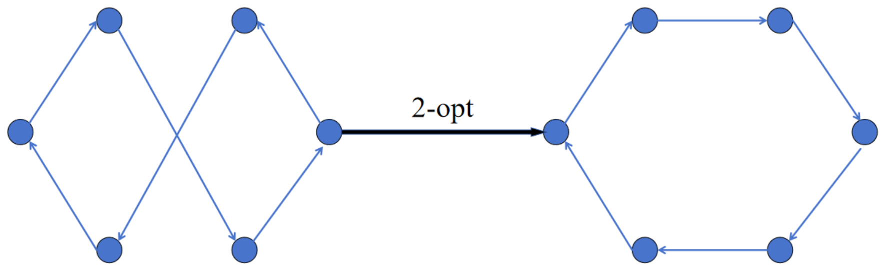


Figure 7. Operational mechanism of the 2-opt local search heuristic for path refinement.

Feeder positions are arranged symmetrically around the feeder slot center (Table 2).

Table 2. Feeder placement on feeder slots for front and rear arms.

Component type	Front arm				Rear arm	
	1	2	3	4	1	2
Components	1,12,14	2	3,6	11,13	4,7,9	5,8,10
Average x-coordinate (mm)	1.60	7.00	4.50	6.50	4.60	2.30
Feeder coordinates (mm)	(2.00,1.00)	(7.00,1.00)	(4.00,1.00)	(6.00,1.00)	(4.00,1.00)	(3.00,1.00)

4.4.3. Evolutionary mechanism for task coordination

Genetic operators tailored for double-chain chromosomes: (1) **Selection:** Tournament selection with a tournament size of three is adopted to preserve superior individuals and ensure population diversity. (2) **Crossover:** Segment exchange is conducted to generate the final offspring chromosome, which is clearly displayed in Figure 8. (3) **Mutation:** Based on the final offspring obtained by crossover, the mutation operation is further implemented to generate the final evolved chromosome, as shown in Figure 9.

The hybrid GA-ACO algorithm leverages the global search capability of the GA and the positive feedback mechanism of ACO to enhance both convergence speed and solution optimality. Whereas the GA optimizes the high-level intercycle sequences to narrow the search space, the ACO utilizes pheromone accumulation to fine-tune intracycle paths. This synergistic approach, integrated with the 2-opt local search, ensures that the algorithm efficiently escapes local optima and converges to a high-quality solution for synchronized dual-arm scheduling.

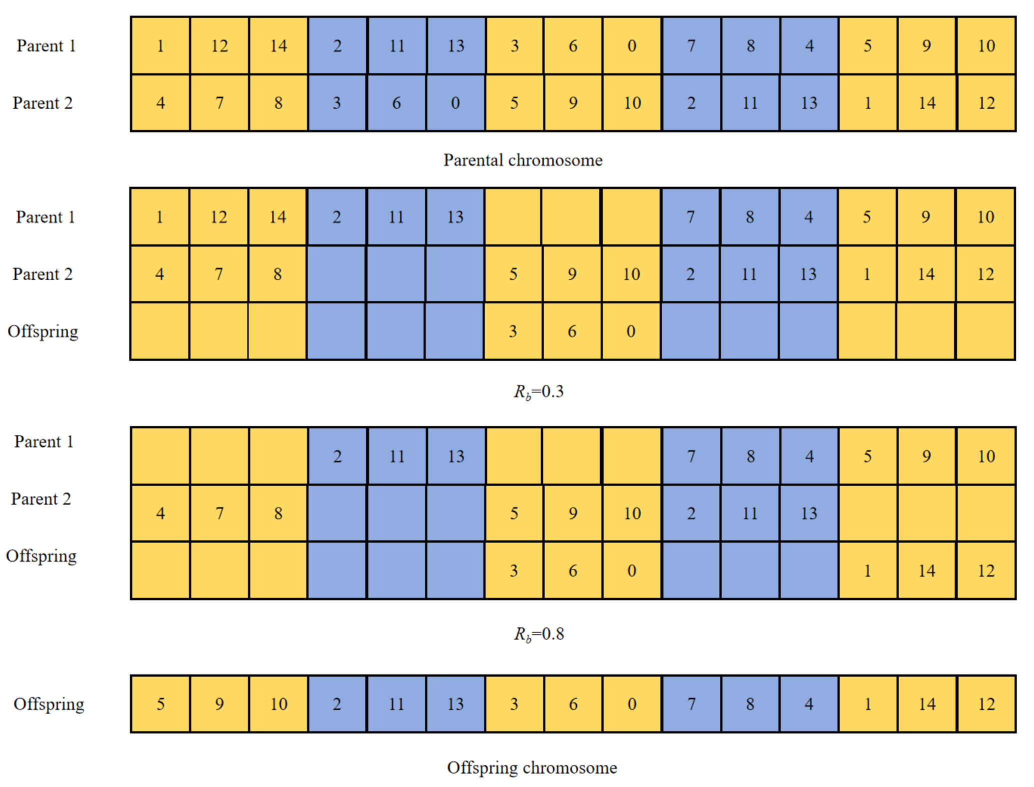


Figure 8. Segment-exchange crossover operator for double-chain chromosomal evolution.

Figure 9. Position-exchange mutation operator for search space exploration.

5. Experimental results and analysis

Based on the preceding mathematical model and hierarchical optimization framework, this section presents simulation experiments to evaluate the proposed HAGA's effectiveness, robustness, and scalability, using real PCB assembly line industrial data. Experiments include representative case validation,and generalization assessment across problem scales.

5.1. Experimental setup and parameter configuration

The experimental setup fully considers the heterogeneous capabilities, nozzle compatibility constraints, and multitype component distribution of placement machines to restore a real manufacturing environment.

5.1.1. Machine specifications

The production line comprises three heterogeneous dual-arm placement machines in series, with key parameters detailed in Table 3. The "Left stopper coordinate" serves as the global reference origin; all local component coordinates (x, y) are transformed to global machine coordinates $(x + 300, y + 350)$ for trajectory calculation.

Table 3. Machine-related parameter configurations for the three heterogeneous placement machines.

Parameter	Machine 1	Machine 2	Machine 3
Number of placement heads	12	8	3
Component picking time (s/chip)	0.12	0.18	0.24
Component placing time (s/chip)	0.12	0.18	0.24
Distance between heads (mm)	8.00	16.00	16.00
Distance between feeder slots (mm)	8.00	16.00	16.00
Average arm speed (mm s^{-1})	800.00	750.00	600.00
Nozzle replacement time (s)	1.50	1.50	1.50
Z-axis movement time (s)	0.13	0.13	0.13
Left stopper coordinate (mm)	(300.00, 350.00)	(300.00, 350.00)	(300.00, 350.00)

5.1.2. PCB and component data

The primary case study uses real production data of a complex PCB with 237 components (28 types, 6 nozzle configurations). Component types, nozzle requirements, and machine compatibility are listed in Table 4; machine-nozzle compatibility constraints are in Table 5; the spatial distribution of all components on the 250.00 mm × 150.00 mm PCB is visually presented in Figure 10 (coordinate transformation: $x_{\text{global}} = x + 300$, $y_{\text{global}} = y + 350$).

Table 4. Relevant data of component placement points including type, nozzle requirements, and quantity.

Type	Nozzles	Machines	Quantity	Type	Nozzles	Machines	Quantity
CT1	130, 1002	2, 3	13	CT15	115, 1001	1, 2	4
CT2	130, 1002	2	4	CT16	115, 1001	1, 2	8
CT3	130, 1002	2	4	CT17	115, 120	1, 2	11
CT4	120	1	4	CT18	115, 1001	1, 2	4
CT5	115, 1001	1, 2	24	CT19	115, 120, 1001	1, 2	8
CT6	120	1	14	CT20	120, 1001	1	4
CT7	115, 120, 1001	1, 2	30	CT21	115, 120	1, 2	8
CT8	115, 120, 1001	1, 2	18	CT22	115	1, 2	6
CT9	130, 240, 1002	2	4	CT23	115	1, 2	4
CT10	115	1, 2	10	CT24	115, 1001	1, 2	4
CT11	130, 240, 1002	2, 3	9	CT25	120, 1002	1, 2	4
CT12	130, 240, 1002	2, 3	2	CT26	1002, 1003	3	6
CT13	115, 120, 1001	1, 2	4	CT27	1002, 1003	3	4
CT14	115, 120, 1001	1, 2	18	CT28	1002, 1003	3	4

Table 5. Machine–nozzle compatibility constraints for the production line.

Machine	Available nozzle types
Machine 1	115, 120, 1001
Machine 2	115, 130, 240, 1002
Machine 3	1002, 1003

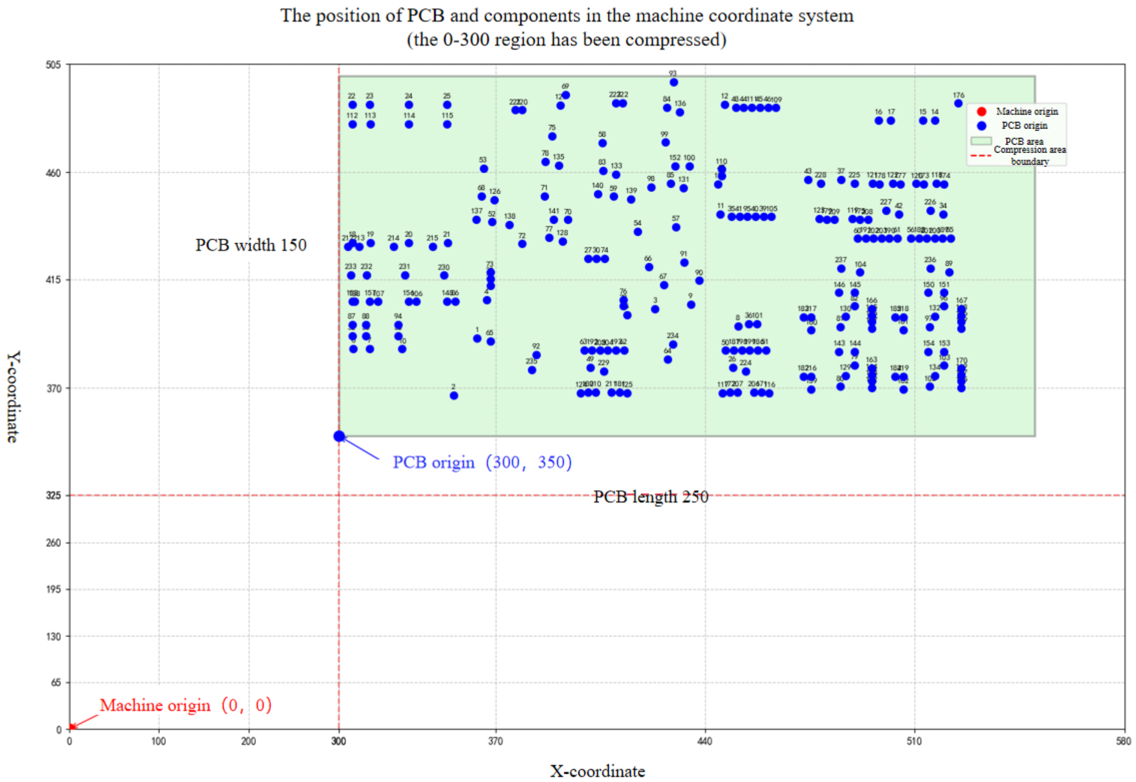


Figure 10. Spatial distribution of component placement positions on the PCB (mm).

5.2. Performance analysis on representative case

5.2.1. Global load balancing (Stage 1)

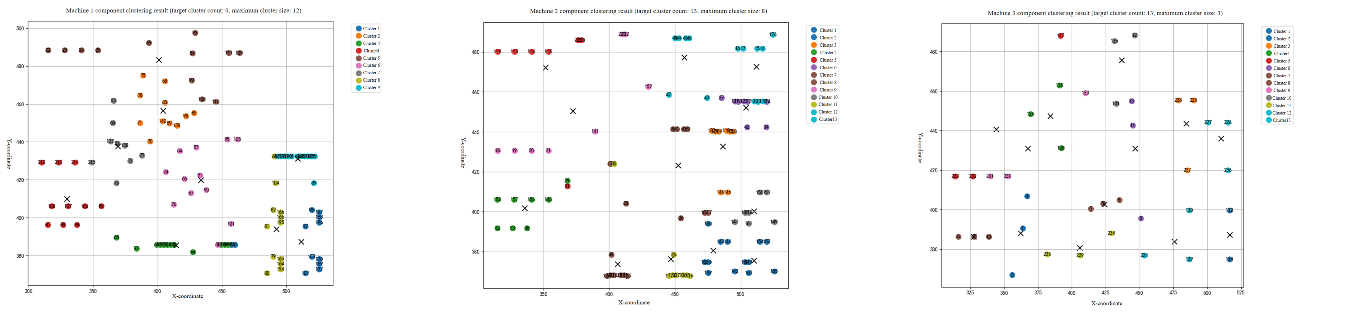
The IAGA distributes component types across machines to balance workload, with results in Table 6. Despite component count differences (102 vs. 38), production times are balanced (81.88 s, 80.59 s, 72.42 s), as Machine 3 handles larger, slower-processed components with fewer heads, confirming the IAGA's consideration of heterogeneous machine capabilities.

Table 6. Statistics of component-to-machine allocation and theoretical production time.

Machine	Components	Time (s)	Component types
Machine 1	102	81.88	4, 6, 7, 8, 13, 16, 19, 20, 21, 23
Machine 2	97	80.59	2, 3, 5, 9, 10, 14, 15, 17, 18, 22, 24, 25
Machine 3	38	72.42	1, 11, 12, 26, 27, 28

5.2.2. Component grouping and clustering (Stage 2)

Improved spectral clustering groups components for each machine, with Machine 1 converging to weights: $w_{\text{spatial}} = 0.6885$, $w_{\text{nt}} = 0.2143$, $w_{\text{ct}} = 0.0972$ (spatial proximity as primary driver). The clustering results of all three machines are displayed in Figure 11, which demonstrates strong spatial compactness for efficient simultaneous picking operations.



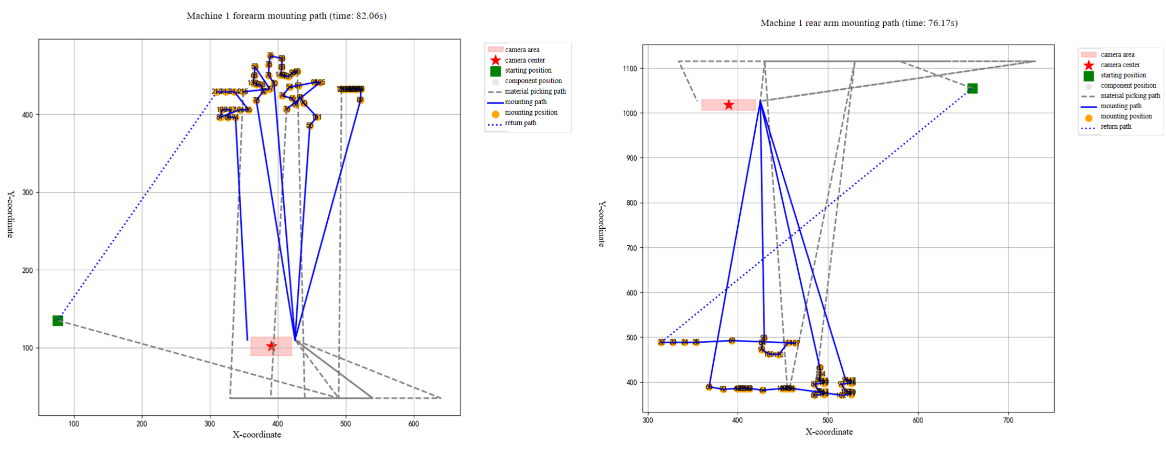
(a) Machine 1 **(b)** Machine 2 **(c)** Machine 3

Figure 11. Clustering results via improved spectral algorithm for heterogeneous machine workloads: (a) Machine 1, (b) Machine 2, (c) Machine 3.

5.2.3. Trajectory optimization (Stage 3)

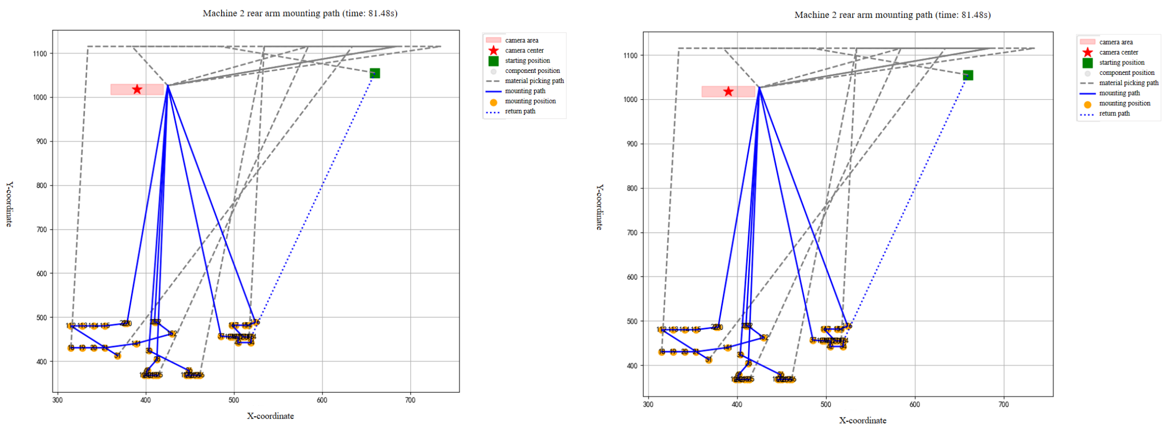
The optimized trajectories of Machines 1 and 2 present regular S-shaped or U-shaped scanning paths, with no cross or collision between dual arms. The backtracking distance is significantly reduced by the ACO with 2-opt local search. The trajectory density matches the component distribution, and feeder positions are optimized by the minimum-distance principle. Machine 3 shows sparser trajectories due to fewer components, whereas the makespan remains balanced, verifying the effectiveness of global load balancing.

Figures 12–14 show the optimized placement trajectories of the front and rear arms for each machine.



(a) Front arm **(b)** Rear arm

Figure 12. Optimized motion trajectories for Machine 1: (a) front arm, (b) rear arm (mm).



(a) Front arm **(b)** Rear arm

Figure 13. Optimized motion trajectories for Machine 2: (a) front arm, (b) rear arm (mm).

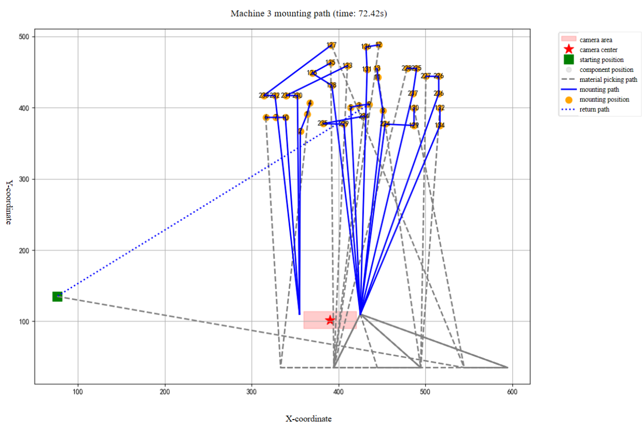


Figure 14. Optimized motion trajectories for front and rear arms of Machine 3 (mm).

5.3. Comparative analysis and generalization verification

Comparative experiments use four datasets (small: Groups 1 and 2; medium: Group 3; large: Group 4) with details in Tables 7–10, benchmarking the HAGA against the K-means GA and the traditional GA.

Table 7. Experimental data of Group 1 for algorithm validation and scalability testing.

Component type	Available nozzle types	Available machines	Placement quantity
CT1	120	1	21
CT2	115, 120, 1001	1, 2	9
CT3	115, 120, 1001	1, 2	9
CT4	120, 1002	1, 2	17
CT5	130, 240, 1002	2	30
CT6	130, 240, 1002	2	8

Table 8. Experimental data of Group 2 for algorithm validation and scalability testing.

Component type	Available nozzle types	Available machines	Placement quantity
CT1	120	1	21
CT2	115, 120, 1001	1, 2	9
CT3	115, 120, 1001	1, 2	9
CT4	120, 1002	1, 2	17
CT5	130, 240, 1002	2	30
CT6	130, 240, 1002	2	8
CT7	1002, 1003	2	3

Table 9. Experimental data of Group 3 for algorithm validation and scalability testing.

Component type	Available nozzle types	Available machines	Placement quantity
CT1	120, 1001	1	43
CT2	115, 120, 1001	1, 2	67
CT3	120	1	40
CT4	130, 1002	2	40
CT5	130, 240, 1002	2, 3	30
CT6	130, 1002	2	8
CT7	1002, 1003	3	2
CT8	1002, 1003	3	2
CT9	1002, 1003	3	2
CT10	130, 240, 1002	2, 3	2
CT11	130, 240, 1002	2, 3	2
CT12	1002, 1003	3	9

Table 10. Experimental data of Group 4 for algorithm validation and scalability testing.

Component type	Available nozzle types	Available machines	Placement quantity
CT1	120	1	97
CT2	130, 240, 1002	2	8
CT3	130, 240, 1002	2, 3	5
CT4	130, 240, 1002	2, 3	7
CT5	130, 240, 1002	2, 3	8
CT6	115, 120, 1001	1, 2	133
CT7	130, 240, 1002	2, 3	8
CT8	130, 240, 1002	2, 3	8
CT9	130, 240, 1002	2, 3	8
CT10	130, 240, 1002	2, 3	8
CT11	130, 240, 1002	2, 3	11
CT12	1002, 1003	3	6
CT13	1002, 1003	3	2
CT14	1002, 1003	3	4
CT15	1002, 1003	3	2
CT16	1002, 1003	3	4
CT17	1002, 1003	3	3

Table 11 shows the HAGA achieves a 12% average time reduction vs. the K-means GA and 15% vs. the traditional GA, with widening performance gaps for larger problem scales, confirming superior scalability. It should be noted that, driven by the urgent practical requirement of our industrial partner to maximize throughput, the primary evaluation criterion in these comparative experiments is strictly confined to the final production time, which directly reflects the factory's core bottleneck.

Table 11. Comparison of total production time (s) between the HAGA, K-means GA, and traditional GA across four datasets.

Algorithm	Group 1	Group 2	Group 3	Group 4
Proposed algorithm	**23.65**	**21.44**	**71.69**	**123.07**
K-means GA	26.17	23.72	80.54	138.43
Traditional GA	28.03	24.98	85.21	146.15

5.4. Sensitivity analysis of penalty coefficients

The Sensitivity analysis of penalty coefficients W_α (nozzle changes) and W_β (pickup cycles) involves 30 combinations ($W_\alpha \in [60, 260], W_\beta \in [5, 35]$). Figures 15 and 16 show the saturation effect of the two penalty coefficients on the production makespan, respectively. Figures 15 and 16 show diminishing returns: W_α saturates at 180, and W_β at 15.

The optimal penalty coefficients $W_\alpha = 200$ (for nozzle changes) and $W_\beta = 15$ (for pickup cycles) are determined based on the saturation effect observed in the sensitivity analysis. When W_α exceeds 180, and W_β exceeds 15, the improvement in makespan becomes insignificant, whereas further increasing the parameters may lead to unstable convergence and unnecessary computational cost. Thus, the selected values achieve the best trade-off between optimization performance and efficiency.

The optimal ranges of $W_\alpha \in [180, 220]$ for nozzle changes and $W_\beta \in [15, 20]$ for pickup cycles provide a direct engineering reference for algorithm parameter configuration in practical production, significantly reducing the time cost associated with parameter tuning. In this study, $W_\alpha = 200$ and $W_\beta = 15$ are adopted as the final parameters.

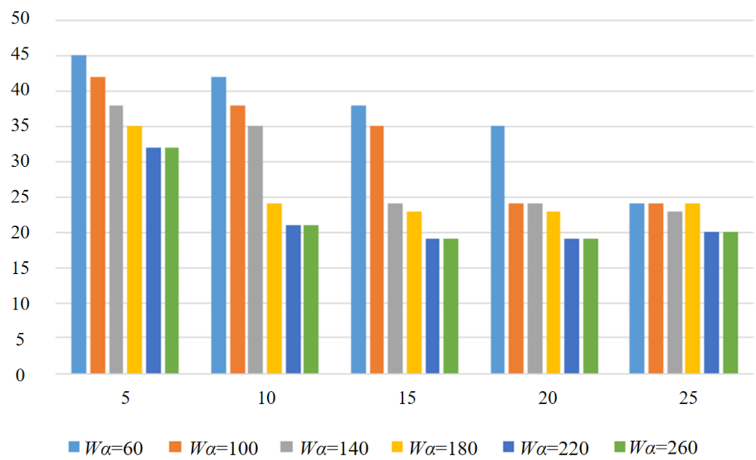


Figure 15. Sensitivity analysis: Influence of penalty coefficient W_α on production makespan.

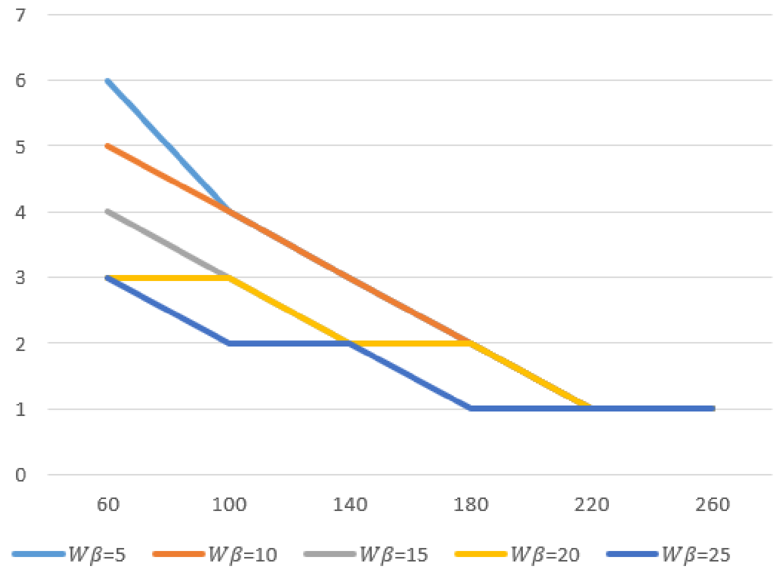


Figure 16. Sensitivity analysis: Influence of penalty coefficient W_β on production makespan.

6. Conclusions

This research conceptualizes multimachine dual-arm PCB placement scheduling as a complex network optimization problem. By proposing a hierarchical decoupling framework based on a HAGA, the study effectively transforms high-dimensional coupling into manageable topological subtasks. By interpreting the scheduling task through the lens of complex network optimization, this research effectively addresses the structural heterogeneity of multimachine systems, ensuring both global system stability and local execution precision. The framework integrates an IAGA for global network balancing, improved spectral clustering for node grouping, and a GA-ACO for synchronized path evolution, ensuring synergistic optimization from global network equilibrium to local machine precision. Experimental results using real-world industrial data demonstrate that the HAGA achieves at least a 10% reduction in the production cycle compared to traditional heuristics. The algorithm exhibits robust stability and scalability, proving its effectiveness in resolving intricate network interdependencies within dual-arm systems.

Despite its promising results, this work has certain limitations. First, the model assumes deterministic processing times and ideal interference-free dual-arm operation, which may slightly differ from dynamic disturbances in real production. Second, the performance of the proposed algorithm is somewhat sensitive to parameter settings, requiring appropriate tuning for different problem scales.

Future research will leverage the heterogeneous dynamical characteristics of the PSN to construct a multiobjective complex network optimization model that simultaneously accounts for production cycles, energy consumption, and equipment wear. By integrating multiobjective evolutionary algorithms with network theory, we aim to enhance the adaptability of the proposed framework in high-mix, low-volume intelligent manufacturing scenarios, thereby further improving the flexibility and cost-effectiveness of smart production systems in complex environments.

Use of AI tools declaration

The authors declare that no artificial intelligence tools were employed throughout the whole process of this study, including research design, mathematical modeling, algorithm development, numerical simulation, data analysis, manuscript drafting and revision. All intellectual work presented in this paper was completed independently by the authors.

Conflict of interest

The authors have no relevant financial or non-financial competing interests to disclose. There are no conflicts of interest that could affect the objectivity and integrity of this research work and the publication of this article.

Acknowledgment

This research is supported by the industrial technical cooperation project with Dalian Rijia Co., Ltd. The authors sincerely appreciate Dalian Rijia Co., Ltd. for providing real production line data, equipment parameters and on-site technical support, which lay a solid foundation for the implementation of experiments. We also thank the editors and anonymous reviewers for their insightful comments and valuable suggestions, which have significantly improved the quality of this manuscript.

Author contributions

Chang Kou and Jihong Shen conceived the research and determined the overall technical framework. Mingze Sun, Ting Song and Jihong Shen established the mathematical model and designed the hierarchical decoupling framework as well as hybrid optimization algorithms. Chang Kou and Mingze Sun implemented the algorithms, conducted simulation experiments and collected production data. Mingze Sun and Ting Song finished data sorting, result analysis and parameter sensitivity analysis. Chang Kou and Mingze Sun wrote the original manuscript. Jihong Shen and Ting Song revised and polished the full paper. Jihong Shen supervised the entire research and managed the project. All authors reviewed the manuscript and approved the final version.

All authors have read and agreed to the published version of the manuscript.

References

1. M. O. Ball, M. J. Magazine, Sequencing of insertions in printed circuit board assembly, *Oper. Res.*, **36** (1988), 192–201. https://doi.org/10.1287/opre.36.2.192

2. G. Y. Zhu, W. B. Zhang, An improved shuffled frog-leaping algorithm to optimize component pick-and-place sequencing optimization problem, *Expert Syst. Appl.*, **41** (2014), 6818–6829. https://doi.org/10.1016/j.eswa.2014.04.038

3. D. Li, S. W. Yoon, PCB assembly optimization in a single gantry high-speed rotary-head collect-and-place machine, *Int. J. Adv. Manuf. Technol.*, **88** (2017), 2819–2834. https://doi.org/10.1007/s00170-016-8942-4

4. H. Y. Lin, C. J. Lin, M. L. Huang, Optimization of printed circuit board component placement using an efficient hybrid genetic algorithm, *Appl. Intell.*, **45** (2016), 622–637. https://doi.org/10.1007/s10489-016-0775-1

5. H. P. Hsu, Solving feeder assignment and component sequencing problems for printed circuit board assembly using particle swarm optimization, *IEEE Trans. Autom. Sci. Eng.*, **14** (2017), 881–893. https://doi.org/10.1109/TASE.2016.2622253

6. A. R. Alazzam, Using BUA algorithm to solve a sequential pick and place problem, in *2018 International Conference on Information and Computer Technologies (ICICT)*, DeKalb, IL, USA, 144–149. https://doi.org/10.1109/INFOCT.2018.8356858

7. J. Mumtaz, Z. Guan, L. Yue, L. Zhang, C. He, Hybrid spider monkey optimization algorithm for multi-level planning and scheduling problems of assembly lines, *Int. J. Prod. Res.*, **58** (2020), 6252–6267. https://doi.org/10.1080/00207543.2019.1675917

8. C. J. Lin, C. H. Lin, Component placement process optimization for multi-head surface mounting machine using a hybrid algorithm, *Sens. Mater.*, **33** (2021), 3529–3545. https://doi.org/10.18494/SAM.2021.3579

9. Z. Li, X. Yu, J. Qiu, H. Gao, Cell division genetic algorithm for component allocation optimization in multifunctional placers, *IEEE Trans. Ind. Inf.*, **18** (2022), 559–570. https://doi.org/10.1109/TII.2021.3069459

10. Z. Li, H. Sun, X. Yu, S. Wei, Heuristic sequencing hopfield neural network for pick-and-place location routing in multi-functional placers, *Neurocomputing*, **472** (2022), 35–44. https://doi.org/10.1016/j.neucom.2021.11.040

11. T. He, D. B. Li, S. W. Yoon, A multi-phase planning heuristic for a dual-delivery SMT placement machine optimization, *Robot. Comput.-Integr. Manuf.*, **47** (2017), 85–94. https://doi.org/10.1016/j.rcim.2016.11.006

12. T. He, D. Li, S. W. Yoon, An adaptive clustering-based genetic algorithm for the dual-gantry pick-and-place machine optimization, *Adv. Eng. Inf.*, **37** (2018), 66–78. https://doi.org/10.1016/j.aei.2018.04.007

13. D. Li, R. Qiang, S. W. Yoon, Balanced adaptive tabu search algorithm to optimize dual-gantry pick-and-place assembly, *Procedia Manuf.*, **11** (2017), 1892–1899. https://doi.org/10.1016/j.promfg.2017.07.331

14. D. S. Sun, T. E. Lee, K. H. Kim, Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool, *Int. J. Prod. Econ.*, **95** (2005), 245–264. https://doi.org/10.1016/j.ijpe.2004.01.003

15. A. Tóth, T. Knuutila, O. S. Nevalainen, Machine configuration and workload balancing of modular placement machines in multi-product PCB assembly, *Int. J. Comput. Integr. Manuf.*, **31** (2018), 815–830. https://doi.org/10.1080/0951192X.2018.1429667

16. Y. Crama, J. Klundert, F. C. R. Spieksma, Production planning problems in printed circuit board assembly, *Discrete Appl. Math.*, **123** (2002), 339–361. https://doi.org/10.1016/S0166-218X(01)00345-6

17. P. Ji, Y. F. Wan, Planning for printed circuit board assembly: The state-of-the-art review, *Int. J. Comput. Appl. Technol.*, **14** (2001), 136–144. https://doi.org/10.1504/IJCAT.2001.000268

18. P. Rogers, R. Warrington, Production planning for surface mount technology lines, *Int. J. Prod. Res.*, **42** (2004), 2693–2718. https://doi.org/10.1080/00207540410001671697

19. S. Emet, T. Knuutila, E. Alhoniemi, M. Maier, M. Johnsson, O. S. Nevalainen, Workload balancing in printed circuit board assembly, *Int. J. Adv. Manuf. Technol.*, **50** (2010), 1175–1182. https://doi.org/10.1007/s00170-010-2558-x

20. S. Rajakumar, V. P. Arunachalam, V. Selladurai, Workflow balancing in parallel machines through genetic algorithm, *Int. J. Adv. Manuf. Technol.*, **33** (2007), 1212–1221. https://doi.org/10.1007/s00170-006-0553-z

21. M. T. Sze, P. Ji, W. B. Lee, Modeling the component assignment problem in PCB assembly, *Assem. Autom.*, **21** (2001), 55–60. https://doi.org/10.1108/01445150110381745

22. L. P. Khoo, D. Alisantoso, Line balancing of PCB assembly line using immune algorithms, *Eng. Comput.*, **19** (2003), 92–100. https://doi.org/10.1007/s00366-003-0256-0

23. G. Zhang, Z. Li, X. Du, A hybrid genetic algorithm to optimize the printed circuit board assembly process, in *International Conference on Logistics Systems and Intelligent Management*, 2010, 563–567. https://doi.org/10.1109/ICLSIM.2010.5461358

24. Y. Wu, P. Ji, A solution method for the component allocation problem in printed circuit board assembly, *Assem. Autom.*, **30** (2010), 155–163. https://doi.org/10.1108/01445151011029790

25. A. Rong, A. Tóth, O. S. Nevalainen, T. Knuutila, R. Lahdelma, Modeling the machine configuration and line-balancing problem of a PCB assembly line with modular placement machines, *Int. J. Adv. Manuf. Technol.*, **54** (2011), 349–360. https://doi.org/10.1007/s00170-010-2920-z

26. T. He, D. B. Li, S. W. Yoon, A heuristic algorithm to balance workloads of high-speed SMT machines in a PCB assembly line, *Procedia Manuf.*, **11** (2017), 1790–1797. https://doi.org/10.1016/j.promfg.2017.07.316

27. J. Mumtaz, Z. Guan, L. Yue, Z. Wang, S. Ullah, M. Rauf, Multi-level planning and scheduling for parallel PCB assembly lines using hybrid spider monkey optimization approach, *IEEE Access*, **7** (2019), 18685–18700. https://doi.org/10.1109/ACCESS.2019.2895954

28. X. Yan, H. Zuo, C. Hu, W. Gong, V. S. Sheng, Load optimization scheduling of chip mounter based on hybrid adaptive optimization algorithm, *Complex Syst. Model. Simul.*, **3** (2023), 1–11. https://doi.org/10.23919/CSMS.2022.0026

29. T. L. Chen, J. C. Chen, Y. J. Chang, The optimal configuration for various placement machines in PCB assembly lines, *Ann. Oper. Res.*, **349** (2025), 365—396. https://doi.org/10.1007/s10479-024-05828-6

30. J. P. Huang, Q. K. Pan, P. N. Suganthan, Effective constructive and composite heuristics for grouping printed circuit boards in the electronic assembly industry, *Eng. Optim.*, **54** (2022), 1722–1758. https://doi.org/10.1080/0305215X.2021.1983806

31. Y. Chen, J. Zhong, J. Mumtaz, S. Zhou, L. Zhu, An improved spider monkey optimization algorithm for multi-objective planning and scheduling problems of PCB assembly lines, *Expert Syst. Appl.*, **229** (2023), 120600 https://doi.org/10.1016/j.eswa.2023.120600

32. H. Gao, Z. Li, X. Yu, J. Qiu, Hierarchical multi-objective heuristic for PCB assembly optimization in a beam-head surface mounter, *IEEE Trans. Cybern.*, **52** (2021), 6911–6924. https://doi.org/10.1109/TCYB.2020.3040788

33. J. Zhong, Y. Chen, J. Mumtaz, A multi-objective scheduling optimization method for PCB assembly lines based on the improved spider monkey algorithm, *Eng. Proc.*, **23** (2022), 15. https://doi.org/10.3390/engproc2022023015

34. J. Du, J. Mumtaz, W. Zhao, J. Huang, Flexsim-simulated PCB assembly line optimization using deep Q-network, *Eng. Proc.*, **75** (2024), 34. https://doi.org/10.3390/engproc2024075034

35. A. Tóth, T. Knuutila, O. S. Nevalainen, A divide and combine method for machine configuration and workload balancing problem in multiple product PCB assembly, *Int. J. Adv. Manuf. Technol.*, **120** (2022), 4079–4095. https://doi.org/10.1007/s00170-022-08819-8