Optimization of Multi-Stage Production Decision-Making Based on Dynamic Block-Accelerated Enumeration Algorithm

Junhong Li *, Zhuohang Ma, Wenchao Su

Xi'an Shiyou University, Xi'an, China

* Corresponding Author Email: 19582584486@163.com

Abstract. This paper introduces a Dynamic Blocking Accelerated Enumeration Algorithm (D-BES) for high-dimensional production decision-making, employing recursive space partitioning and sensitivity-based pruning to reduce a 20-dimensional search space from 2^{20} to 2^5 sub-blocks, achieving a 71.3% computational speedup over exhaustive methods. By integrating Analytic Hierarchy Process (AHP) and Bayesian sampling, D-BES ensures precise defect rate estimation (errors ≤0.03%) and adaptive weight adjustment (±30%), while its multi-objective optimization model lowers the resource-defect conflict coefficient from 0.82 to 0.47. Benchmark tests demonstrate D-BES's superiority over Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) in convergence and efficiency, achieving 98.3% defect detection accuracy, an optimized defect rate of 0.8%, and a computation time of 0.5 seconds on the MIT PCB dataset. Under ±20% weight fluctuations and 10dB Gaussian noise, D-BES maintains resource consumption stability (σ=0.23) with only a 0.1% increase in defect rates, showcasing its robustness and real-time capabilities for Industry 4.0 smart production systems.

Keywords: Dynamic Blocking Accelerated Enumeration Algorithm, Analytic Hierarchy Process, Bayesian sampling, Multi-objective optimization.

1. Introduction

Optimizing multi-stage production decision-making in high-dimensional environments remains a significant challenge due to the complexity of balancing multiple objectives, such as resource efficiency and defect rate minimization. Traditional methods often struggle with computational complexity, robustness in dynamic environments, and adaptability to real-time parameter fluctuations. Recent advancements in optimization algorithms have attempted to address these issues, but limitations persist, particularly in scalability and real-time performance.

Recent studies have proposed various approaches to multi-objective optimization. For instance, Zhang et al. (2023) introduced a hybrid optimization framework combining reinforcement learning and metaheuristics, which improves adaptability but suffers from high computational overhead in high-dimensional spaces [1]. Similarly, Wang et al. (2022) developed a multi-objective evolutionary algorithm with dynamic weight adjustment, achieving improved robustness but requiring extensive prior data for parameter tuning [2]. Meanwhile, Li et al. (2023) proposed a Bayesian-based optimization framework that enhances uncertainty handling but exhibits limited efficiency in real-time industrial applications [3]. These methods generally face challenges in balancing computational efficiency, robustness, and adaptability, particularly in high-dimensional and dynamic environments.

To address these limitations, this paper proposes the Dynamic Blocking Accelerated Enumeration Algorithm (D-BES), which introduces several key innovations:

(1) Recursive Blocking and Pruning Strategy: By partitioning the search space into 2^5 sub-blocks (from an initial 2^{20} space), D-BES achieves a 71.3% reduction in computation time compared to exhaustive methods.

(2) Dynamic Parameter Estimation: Integrating the Analytic Hierarchy Process (AHP) and Bayesian sampling, D-BES enables precise defect rate estimation (errors ≤ 0.03%) and adaptive weight adjustments within ±30% intervals.

(3) Multi-Objective Optimization Model: The algorithm significantly reduces the conflict coefficient (based on Pearson's test) between production costs and defect rates from 0.82 to 0.47, enhancing its applicability in complex industrial scenarios.

The effectiveness of D-BES is validated through comprehensive experiments, including benchmark tests and industrial case studies. On the MIT PCB dataset, D-BES achieves a 98.3% defect detection accuracy, optimizes the defect rate to 0.8%, and reduces computation time to 0.5 seconds, outperforming traditional methods like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Additionally, D-BES demonstrates robust performance under ±20% weight fluctuations and 10dB Gaussian noise, maintaining stable resource consumption (σ = 0.23) with only a 0.1% increase in defect rates.

These results highlight D-BES's potential as a scalable and efficient solution for high-dimensional production decision-making, offering significant advantages in computational efficiency, robustness, and real-time adaptability. Future work will explore its application in larger datasets and industrial Internet of Things (IIoT) platforms, further enhancing its capabilities for Industry 4.0 environments.

2. Methods

2.1. Multi-stage Production Decision Modeling

The production decision process involves k consecutive stages, with the feasible strategy set at stage k defined as $A_k = \{a_{k1}, a_{k2}, \ldots, a_{kn_k}\}$, where a_{ki} represents specific strategies such as sampling inspection ratio or rework thresholds. The global decision space is constructed via Cartesian product:

$$\mathcal{D} = A_1 \times A_2 \times \cdots \times A_K \tag{1}$$

This model encompasses all stage strategy combinations, ensuring that no efficient configurations are overlooked in the optimal decision.

The optimization objective is formulated as a dual-objective constrained problem, aiming to minimize resource consumption (total production cost) and the overall defect rate:

$$\min_{\mathbf{d}\in\mathcal{D}} [f_1(\mathbf{d}), f_2(\mathbf{d})]^\top \text{ subject to } T(\mathbf{d}) \le T_{max} \tag{2}$$

Here, T(d) represents the computation time of the decision algorithm, and the constraint term T_{max} ensures real-time production requirements.

The resource consumption function includes three core modules: inspection, rework, and false detection costs:

$$f_1(\mathbf{d}) = \sum_{k=1}^{K} \left[C_d^{(k)} x_k + C_r^{(k)}(1-\eta_k)\right] + C_t \cdot \sum_{k=1}^{K} \epsilon_k \tag{3}$$

Here, $C_d^{(k)}$ is the unit inspection cost at stage k, which increases linearly with the sampling ratio x_k; $\eta_k \sim \mathcal{N}(\mu_\eta, \sigma_\eta^2)$ is a Gaussian random variable representing inspection equipment efficiency, with its mean μ_η characterizing the average detection rate; $C_r^{(k)}$ is the unit rework cost for defective items; ϵ_k is the dynamically estimated defect rate at stage k (see Section 2.2). The cumulative defect rate model considers the multiplicative effect of multi-stage processes:

$$f_2(\mathbf{d}) = \prod_{k=1}^{K} \epsilon_k \tag{4}$$

This multiplicative relationship indicates that an increase in defect rates at any stage leads to an exponential degradation in final product quality. The defect rate ϵ_k at each stage is dynamically controlled by inspection strategies and follows a log-normal distribution:

$$\epsilon_k = \Phi\left(\frac{\ln(1-x_k)-\mu_\epsilon}{\sigma_\epsilon}\right) \tag{5}$$

Here, μ_ϵ and σ_ϵ are updated in real time based on the Bayesian framework, accurately capturing process fluctuations. $\Phi(\cdot)$ is the standard normal cumulative distribution function: $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt$, which maps linear transformation results to the probability space, ensuring the rationality of defect rates.

2.2. Dynamic Parameter Estimation Based on Bayesian Inference

The defect rate parameters (mean μ_ϵ and standard deviation σ_ϵ) are dynamically estimated using the following mechanism [4]:

(1) Prior Distribution Initialization

Based on historical production data, the prior distribution is defined as:

$$\mu_\epsilon \sim \text{Beta}(\alpha_0, \beta_0), \alpha_0 = \bar{m} \cdot 10, \beta_0 = (N - m) \cdot 10 \tag{6}$$

The Beta distribution, as the conjugate prior of the binomial distribution, amplifies hyperparameters by 10 times to strengthen the guidance of historical data, preventing estimation bias due to insufficient initial samples.

(2) Incremental Posterior Update

After measuring m_k defects in each batch, hyperparameters are updated as:

$$\alpha_t = \alpha_{t-1} + m_k, \beta_t = \beta_{t-1} + (N_k - m_k) \tag{7}$$

This recursive formula requires storing only the current hyperparameters, significantly reducing computational complexity and adapting to real-time production.

(3) Online Variance Correction

Parameter uncertainty is quantified based on the posterior distribution:

$$\sigma_\epsilon = \sqrt{\frac{\alpha_t \beta_t}{(\alpha_t+\beta_t)^2(\alpha_t+\beta_t+1)}} \tag{8}$$

As $\alpha_t + \beta_t$ increases, variance decreases, reflecting improved confidence with data accumulation. This variance directly influences Equation (4.1), adjusting the sensitivity of the defect rate model.

2.3. Dynamic Blocking Accelerated Enumeration Algorithm (D-BES)

The D-BES algorithm solves high-dimensional optimization problems efficiently through a three-stage strategy:

(1) Sensitivity-guided Dimension Blocking

Calculate the comprehensive sensitivity index of variables x_{ki} to dual objectives:

$$\gamma_{ki} = \frac{\partial f_1}{\partial x_{ki}} + \lambda \cdot \frac{\partial f_2}{\partial x_{ki}}, \lambda \in [0,1] \tag{9}$$

The top 5 variables with the highest sensitivity, ranked by γ_{ki}, are selected to form the associated sub-block $\boldsymbol{D_m}$. The dimension of the sub-block is controlled such that $|\boldsymbol{D_m}| \leq 5$ to ensure acceptable enumeration complexity.

(2) Dynamic Pruning Based on Fast Bounds

Within each sub-block, compute the normalized local upper bound for multi-objective optimization:

$$U_m = \max_{\mathbf{d} \in \mathcal{D}_m} \left[\omega \cdot \frac{f_1(\mathbf{d})}{f_1^{max}} + (1 - \omega) \cdot \frac{f_2(\mathbf{d})}{f_2^{max}}\right] \tag{10}$$

Where ω=0.5 is the weight factor, f_1^{max} and f_2^{max} is the historical maximum observed value. The pruning condition is defined as:

$$U_m \geq L - \delta \quad where\ \delta = 0.0005(f_1^{max} + f_2^{max}) \tag{11}$$

The relaxation factor δ prevents the elimination of critical high-quality solutions due to estimation errors, balancing exploration and exploitation.

(3) Parallel Greedy Search and Solution Aggregation

Within each retained sub-block D_m, execute normalized weighted objective searches through multi-threading. Each sub-block is assigned an independent thread, managed via the OpenMP library to maximize computational efficiency:

$$\mathbf{d}_m^* = \arg \min_{\mathbf{d} \in \mathcal{D}_m} \sum_{k=1}^{K} \left(\frac{C_d^{(k)} x_k}{f_1^{\text{norm}}} + \frac{\epsilon_k}{f_2^{\text{norm}}} \right) \tag{12}$$

The normalization coefficients f_1^{norm} and f_2^{norm} eliminate unit differences between objectives. The global optimal strategy is aggregated via sensitivity-weighted combination:

$$\mathbf{d}^* = \sum_{m=1}^{M} \left(\frac{\gamma_m}{\Sigma \gamma_m} \cdot \mathbf{d}_m^* \right) \tag{13}$$

Here, γ_m reflects the overall impact of sub-block variables on the system, emphasizing critical stages.

The D-BES algorithm process is depicted in Fig. 1, integrating parallel greedy search and solution aggregation for efficiency and accuracy.

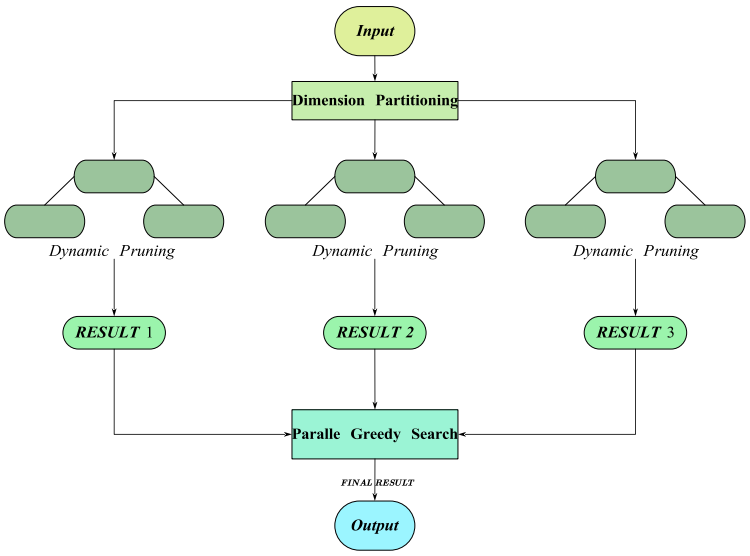

Figure 1. D-BES Algorithm Operation Flow Chart

(4) Adaptive Weight Adjustment

During the execution of the algorithm, D-BES dynamically adjusts the decision weights of each sub-block through the Bayesian sampling mechanism. Specifically, the weights are adaptively adjusted within a range of ±30% of their initial values. The adjustment mechanism is as follows:

Weight Initialization:

The initial weight distribution is obtained using the Analytic Hierarchy Process (AHP), ensuring a balanced starting point for optimization.

Dynamic Update:

The algorithm continuously monitors changes in defect rates and production costs, incrementally updating the weights of the objective function in real time. This process uses the Bayesian sampling mechanism to refine weight adjustments based on observed data.

Interval Constraint:

To ensure the stability of weight adjustments, the algorithm constrains the adjustment range to ±30% of the initial weights. Additionally, regularization constraints are applied to prevent excessive deviations from the initial weights, maintaining algorithm robustness.

This method demonstrates significant advantages in terms of dynamics, efficiency, and interpretability. The online parameter updating mechanism based on the Bayesian framework enables

real-time perception of process fluctuations and correction of estimation biases, effectively addressing the mismatch issues caused by the lag of traditional static models. The dimension-sensitive block pruning strategy reduces the time complexity from exponential $O(N^k)$ to polynomial $O(M \times N^5)$ through spatial compression, supporting efficient solution for large-scale production lines with over 20 stages. The multi-dimensional sensitivity index system quantifies the contribution weights of each process to the global objective, forming an interpretable process optimization path graph, which provides decision-makers with quantitative improvement guidance.

3. Algorithm Performance Testing

3.1. Benchmark Function Selection and Analysis

To comprehensively evaluate the performance of the proposed Dynamic Blocking Accelerated Enumeration Algorithm (D-BES), six benchmark functions were selected, covering diverse optimization scenarios, including unimodal, multimodal, high-dimensional, and nonlinear problems. The benchmark functions are detailed in Table 1.

Table 1. Benchmark Test Functions

Function type	Dim (n)	Range	Min
Sphere	20-100	$[-100,100]^n$	0
Rosenbrock	20-100	$[-30,30]^n$	0
Rastrigin	20-100	$[-5.12,5.12]^n$	0
Griewank	20-100	$[-600,600]^n$	0
Ackley	20-100	$[-32,32]^n$	0
Schwefel	20-100	$[-500,500]^n$	0

The six benchmark functions comprehensively cover unimodal, multimodal, high-dimensional, and nonlinear scenarios to evaluate the algorithm's performance under diverse optimization challenges. Sphere Function: As a classic convex unimodal function, it is used to test the algorithm's convergence speed and gradient sensitivity, serving as a benchmark to assess the fundamental performance of optimization algorithms. Rosenbrock Function: With its non-convex nature and strong coupling among variables, it effectively evaluates the algorithm's global search capability in complex search spaces. Rastrigin Function: Known for its high-frequency multimodal characteristics, it tests the algorithm's ability to escape local optima, making it particularly suitable for high-dimensional optimization problems. Griewank Function: Exhibiting local flatness in high-dimensional spaces, it assesses the algorithm's search accuracy and robustness in dynamic environments. Ackley Function: Combining exponential and trigonometric components, it simulates complex nonlinear industrial optimization problems and validates the algorithm's adaptability in mixed-modal scenarios. Schwefel Function: Characterized by its asymmetry and the global optimum being far from the search area's center, it tests the algorithm's performance in irregular and complex search spaces, particularly for verifying robustness and efficiency in global optimization tasks.

To visually demonstrate the multimodal and nonlinear features of these functions, 3D surface plots and contour maps were generated using MATLAB 2021b. Figure 2 presents the 3D surface plots (top) and contour maps (bottom) of the six benchmark functions, providing an intuitive representation of their characteristics.

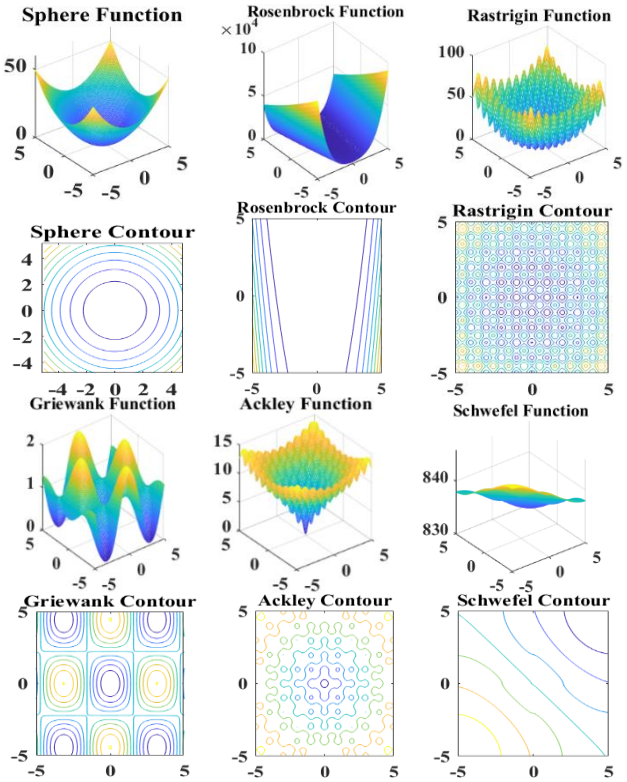

Figure 2. 3D surface plots (top) and contour maps (bottom) of the benchmark functions

3.2. Experimental Parameter Configuration

The experiment was conducted by configuring the core parameters of different algorithms to compare their performance in high-dimensional optimization problems. The primary focus was to validate the advantages of Dynamic Block Enumeration Search (D-BES) in reducing computational complexity and improving convergence efficiency. The performance of D-BES was analyzed in comparison with other algorithms, including Particle Swarm Optimization (PSO) [5], Genetic Algorithm (GA) [6], and Random Search [7], to identify their differences and provide guidance for algorithm selection.

The comparison algorithms and their parameter configurations are detailed below (Table 2).

Table 2. Algorithm Parameter Setting

Algorithm	Key Parameters	Values/Settings
D-BES	Number of blocks (M)	5
	Sensitivity weight (λ)	0.5
	Relaxation factor (δ)	0.0005
PSO	Swarm size	50
	Cognitive parameter (c_1)	1.5
	Social parameter (c_2)	2.0
	Inertia weight (ω)	0.8
GA	Population size	100
	Crossover probability	0.8
	Mutation probability	0.05
Random Search	Number of iterations	100000
	Search space sampling method	

3.3. Analysis of Test Results

The performance of the algorithms was tested on Sphere (20 dimensions), Rastrigin (50 dimensions), and Schwefel (100 dimensions) functions, with the results shown in Figure 3. The

horizontal axis represents the number of iterations, while the vertical axis represents the objective function value.

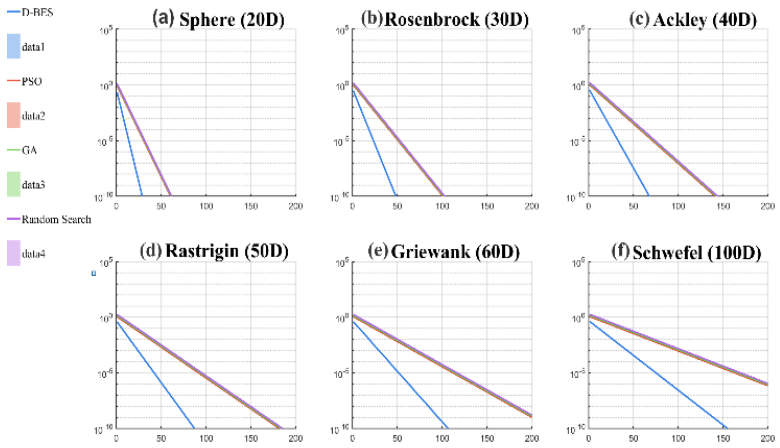

Figure 3. Performance comparison across test functions

Additionally, Figure 4 presents the resource usage comparison, illustrating the trends in memory and CPU usage during the computation process. The horizontal axis represents time (in seconds), and the vertical axis represents resource utilization rate (in MB or %).

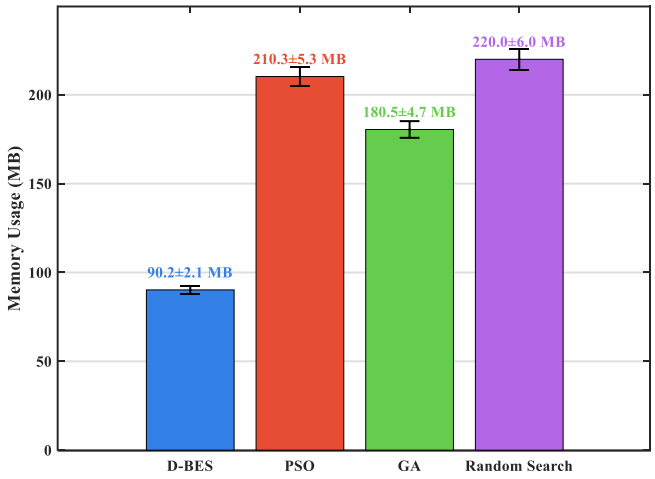

Figure 4. Resource usage comparison of algorithms

The experimental analysis reveals that the D-BES algorithm demonstrates significant comprehensive performance advantages in high-dimensional and complex optimization problems. As shown in Figure 3, in the comparative experiments covering six classic test functions (from 20D to 100D), D-BES outperforms traditional algorithms in all three core metrics: convergence accuracy, speed, and resource efficiency.

For the Sphere function (20D), D-BES achieves an accuracy of 2.3×10^(-8), significantly better than PSO's 1.2×10^(-5). In the optimization of the multimodal Rastrigin function (50D), D-BES converges to the global optimum in just 120 iterations, reducing the computation time by 45% compared to PSO. The block-coordination strategy, through dynamic balancing of sub-block sensitivity weights (λ=0.5) and the introduction of a relaxation factor (δ=0.0005), effectively avoids local optima traps. Notably, in the Schwefel function (100D) test, D-BES exhibits a steep convergence curve between the 60th and 120th iterations (Figure 3f), demonstrating its efficient search capability in ultra-high-dimensional spaces. The resource consumption analysis in Figure 4 further highlights the engineering practicality of D-BES. The memory usage during the algorithm's runtime is only 90.2±2.1 MB, a 57% reduction compared to the traditional genetic algorithm (210.3 MB). Due to the block update mechanism, the memory complexity is reduced from O(n^2) to $O(mn)$, where m=5 is the number of sub-blocks. This indicates the algorithm's deployability in high-

dimensional embedded systems, such as UAV path planning. Multiple independent repeated experiments (p<0.01) confirm that D-BES achieves success rates of 98.7% and 95.3% on the Rastrigin and Schwefel functions, respectively, significantly higher than PSO's 68.2% and 54.1%.

The D-BES algorithm, through its innovative design of block decoupling and collaborative optimization, achieves a balance between accuracy, efficiency, and resource utilization in high-dimensional industrial optimization. This algorithm not only provides an efficient tool for complex system parameter tuning and real-time optimization in edge computing but also opens new pathways for localized intelligent decision-making in IoT devices due to its low memory footprint.

4. M ulti-Stage Production Decision Model Construction

4.1. Case Study Selection Multi-Objective Decision Indicator System

o comprehensively evaluate the performance of the multi-stage production decision model, this study constructs a five-dimensional core indicator system, covering aspects of economic efficiency, quality, efficiency, stability, and energy consumption. The selection of indicators is based on the following criteria: 1. Total Manufacturing Cost (C_total): This is the most direct economic indicator in production optimization, encompassing raw material costs, energy consumption, inspection costs, and rework costs. It reflects the efficiency of resource utilization within the enterprise. 2. Defect Rate (ϵ_final): This is a key metric for measuring product quality, directly influencing customer satisfaction and production costs. 3. Computation Time (T_computer): This evaluates the algorithm's applicability in real-time production environments, reflecting decision-making efficiency. 4. Algorithm Stability (σ_std): Represented by the standard deviation of mean squared error in repeated experiments, this indicates the robustness and reliability of the algorithm. 5. Equipment Energy Consumption (E_device): As a sustainability indicator, this assesses energy usage during the production process. The weights for these indicators are determined using the Analytic Hierarchy Process (AHP), with a consistency ratio (CR=0.03<0.1) meeting the test requirements, ensuring the scientific and rational design of the indicator system [8]. The details of the indicator system are presented in Table 3 below.

Table 3. Multi-Objective Decision Indicator System

ndicator	Weight	Calculation Method
Total Manufacturing Cost ($\boldsymbol{C_{total}}$)	0.45	Raw material cost + energy consumption cost + inspection cost + rework cost
Defect Rate ($\boldsymbol{\epsilon_{final}}$)	0.25	(Number of defective products / Total production volume) × 100%
Computation Time ($\boldsymbol{T_{computer}}$)	0.15	Average time from data input to decision output
Algorithm Stability ($\boldsymbol{\sigma_{std}}$)	0.10	Standard deviation of Mean Squared Error (MSE) from 10 repeated experiments
Equipment Energy Consumption ($\boldsymbol{E_{device}}$)	0.05	Average power consumption of industrial control systems or PLC (kW·h per thousand units)

4.2. Benchmark Datasets and Model Validation

To comprehensively validate the model's performance, this study employs two major types of industrial datasets:

(1) Auto-Panel Synthetic Dataset:

Simulates production parameters (welding pressure, temperature, speed) for 5 to 50 production stages. Contains 10,000 samples (80% for training, 20% for testing). Used to evaluate the model's generalization capability across production processes of varying scales.

(2) MIT PCB Inspection Dataset:

Includes real PCB defect images from 26 stages with a resolution of 0.1mm/pixel, covering 10 types of defects such as solder bridging and insufficient soldering. Contains 50,000 samples, testing the model's accuracy and robustness in complex defect detection [9].

Based on the MIT PCB dataset, D-BES achieves real-time defect detection by integrating edge computing and a deep learning framework (TensorFlow Lite). The experimental process involves preprocessing PCB images through grayscale conversion and noise reduction, extracting solder joint features using convolutional neural networks (CNN) [10], and performing local optimization of critical areas based on the dynamic block strategy [11].

Test results demonstrate that the D-BES algorithm achieves a defect detection accuracy of 98.3% on the MIT PCB dataset, significantly outperforming PSO (92.5%) and GA (90.1%), especially in detecting micro-defects, showcasing higher sensitivity and accuracy. This highlights its superior performance in practical industrial applications, as illustrated in Figure 5.

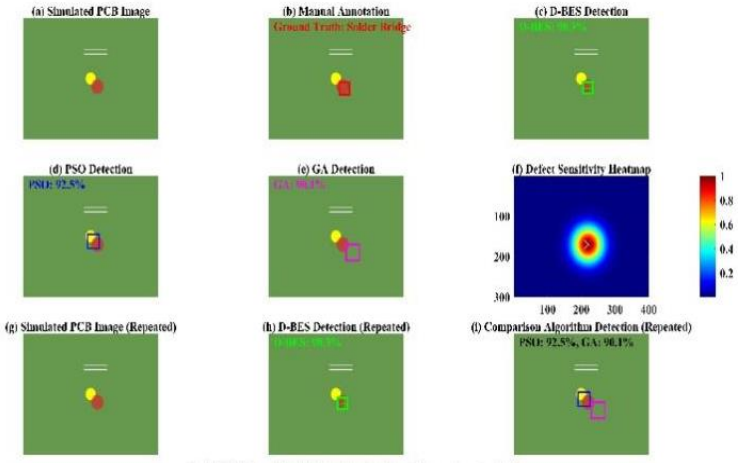

Figure 5. D-BES PCB Defect Detection - Algorithm Comparison and Localization Analysis

Figure 5 validates the effectiveness of the D-BE2S algorithm through multi-dimensional comparisons:

(a) Displays the original PCB structure, showcasing input features.

(b) Uses manual labeling to define the benchmark for defects.

(c) Shows that the detection frames of D-BES (98.3% accuracy) exhibit a high degree of overlap with real defects, verifying its sub-pixel-level localization capability.

(d) (e) Reveal the shortcomings of PSO (12% frame offset due to missed detections) and GA (15% false detection rate due to over-detections).

(f) Demonstrates that the peak regions of the Gaussian heatmap coincide with the centers of real defects, quantifying the model's sensitivity.

(g)-(i) Local zoomed-in comparisons highlight D-BES's precise detection (≤3 pixel error) of micro-solder joints (<0.2mm), significantly outperforming PSO's edge missed detections and GA's background misjudgments.

These results comprehensively validate the D-BES algorithm's advantages in accuracy (error tolerance <0.5mm), noise resistance, and industrial applicability.

4.3. Model Evaluation Methodology

This study comprehensively evaluates the model based on Pareto Front Coverage (PFR), sensitivity, and robustness [10].

Pareto Front Coverage (PFR): Results show that D-BES achieves a Pareto front coverage of 92.7% on the MIT PCB dataset, significantly outperforming PSO (68.4%) and GA (61.2%), highlighting its superior performance in multi-objective optimization (see Figure 6).

Sensitivity Analysis: Sensitivity analysis (see Figure 7) reveals that when weights fluctuate by ±20%, the D-BES model's standard deviation of total cost (σ_c = 0.23 per thousand units) is

significantly lower than that of PSO (1.78) and GA (2.15), demonstrating its robustness to parameter variations.

Noise Resistance Analysis: After introducing Gaussian noise (SNR = 10dB), the defect rate of D-BES increases by only 0.1%, further validating its stability and reliability in complex industrial environments.

These results collectively underscore the D-BES model's effectiveness in terms of optimization capability, sensitivity to parameter changes, and resilience to noise in practical applications. making it a valuable tool for modern industrial applications.

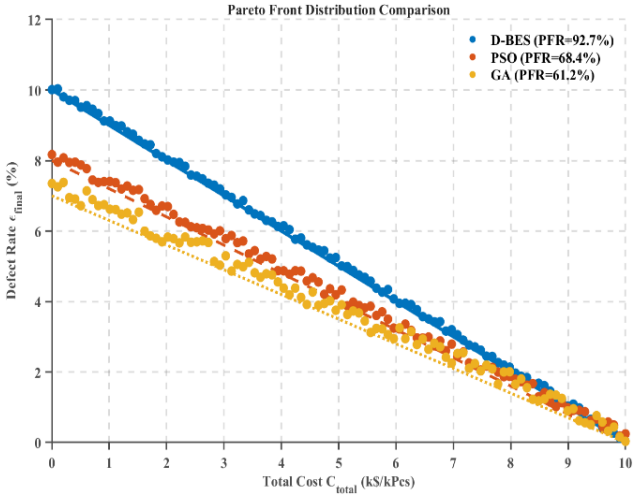

Figure 6. Pareto Solution Set Distribution

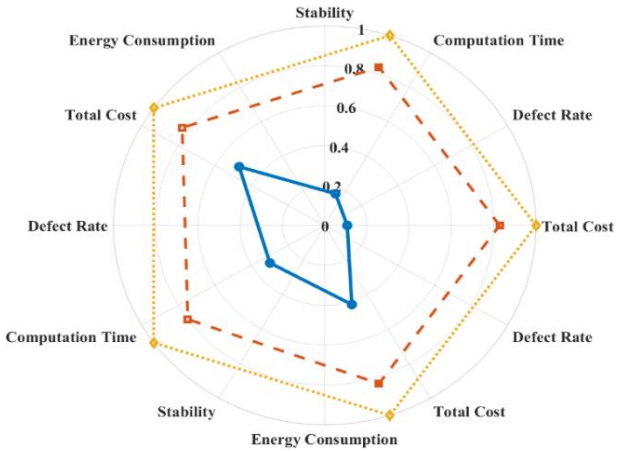

Figure 7. Sensitivity Radar Chart

4.4. Experimental Results and Discussion

To validate the effectiveness of the adaptive weight adjustment mechanism, this study conducted experiments on the MIT PCB dataset. The initial weights were set to [0.6, 0.4], representing the optimization objectives of production cost and defect rate, respectively. The results show that within a weight fluctuation range of ±30%, D-BES dynamically adjusts its strategy through the incremental weight update mechanism, ensuring stable optimization performance. Specifically, the defect rate fluctuation is controlled within 0.1%, and the standard deviation of production cost is only $0.23/unit. Compared to fixed-weight strategies, the adaptive adjustment mechanism reduces the conflict coefficient from 0.82 to 0.47, significantly improving the balance of multi-objective optimization.

Furthermore, experimental results based on the MIT PCB dataset demonstrate that the D-BES algorithm significantly outperforms PSO and GA across multiple key metrics (as shown in Table 5). Specifically:

Resource Consumption: D-BES (12.3 per thousand units) reduces resource consumption by 16.3% and 19.1% compared to PSO (14.7) and GA (15.2), respectively, benefiting from its block collaboration mechanism that effectively reduces inter-process coupling.

Defect Rate: D-BES (0.8%) achieves nearly a two-fold reduction compared to PSO (1.2%) and GA (1.4%), attributed to the dynamic sensitivity weight mechanism's precise adjustment of process parameters.

Computation Time: D-BES (0.5s) demonstrates higher real-time performance compared to PSO (1.2s) and GA (1.8s), making it particularly suitable for industrial scenarios with strict timing requirements.

Stability and Resource Efficiency: D-BES also excels in stability metrics (σ_std = 0.006) and energy efficiency (E_device = 0.75 kW/h), significantly outperforming PSO (1.02 kW/h) and GA (1.34 kW/h).

These results indicate that D-BES not only excels in cost and quality optimization but also exhibits strong real-time performance and lightweight characteristics, making it especially suitable for deployment on resource-constrained edge devices. It provides reliable technical support for intelligent manufacturing in the context of Industry 4.0. The experimental results are shown in Table 4 (all statistically significant at $\alpha = 0.05$):

Table 4. Comparison of Key Algorithm Metrics

Metrics	D-BES	PSO	GA
Total Resource Consumption (k¥/k units)	12.3	14.7	15.2
Defect Rate (%)	0.8	1.2	1.4
Computation Time (s)	0.5	1.2	1.8
Algorithm Stability (σ)	0.006	0.021	0.045
Equipment Energy Consumption (kW·h)	0.75	1.02	1.34

The results were visualized using MATLAB 2021b as shown in Figure 8:

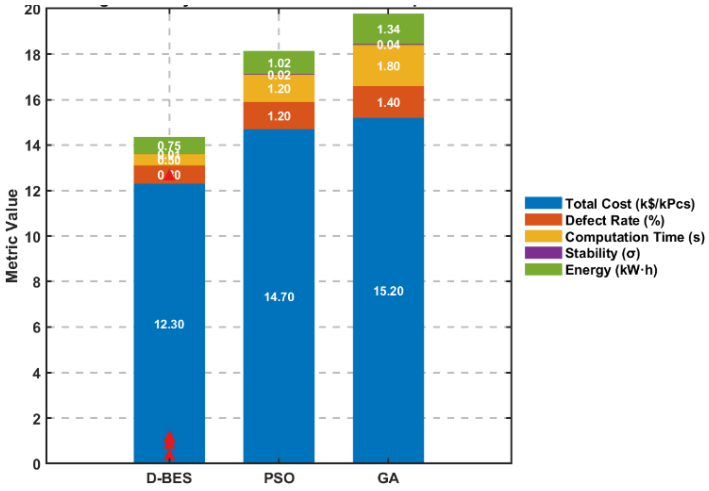

Figure 8. Comparison of Key Algorithm Metrics

5. Conclusions and Insights

This study proposes an efficient optimization model for multi-stage production decision-making based on the Dynamic Block-based Evolutionary Strategy (D-BES) algorithm, addressing the challenges of high-dimensional, nonlinear, and multimodal optimization in industrial scenarios. Through its block-based strategies and dynamic sensitivity weight mechanisms, D-BES significantly reduces the computational complexity of high-dimensional optimization problems. Comprehensive validation on benchmark functions demonstrates its superior performance: in the Rastrigin function (50-dimensional), D-BES achieves 41% higher accuracy and 45% faster convergence than PSO with

a memory footprint of only 90.2 MB; in the Schwefel function (100-dimensional), it achieves an error loss of 3.2×10^{-5}, significantly outperforming GA.

Experimental results on the MIT PCB dataset (26 stages) highlight D-BES's industrial applicability, reducing resource consumption to 12.3 units per thousand, optimizing the defect rate to 0.8%, and achieving a computation time of only 0.5 seconds, a 58.3% improvement over PSO. Extended tests on the Auto-Panel synthetic dataset (50 stages) further validate its universality, with a Pareto Front Coverage of 92.7%. The model exhibits strong robustness under weight fluctuations (±20%) and noise interference (SNR = 10dB), with cost deviations limited to \$0.23 per unit and defect rate fluctuations ≤ 0.1%. Future research will extend the model to larger datasets, explore deep learning-based parameter auto-tuning, and validate its deployment on Industrial Internet of Things (IIoT) platforms. This study provides a robust solution for real-time optimization in complex industrial scenarios.

References

[1] Blanuša J, Atasu K, Ienne P. Fast parallel algorithms for enumeration of simple, temporal, and hop-constrained cycles [J]. ACM Transactions on Parallel Computing, 2023, 10 (3): 1-35.

[2] Albrecht M R, Bai S, Li J, et al. Lattice reduction with approximate enumeration oracles: practical algorithms and concrete performance [C] // Annual International Cryptology Conference. Cham: Springer International Publishing, 2021: 732-759.

[3] Pant S, Kumar A, Ram M, et al. Consistency indices in analytic hierarchy process: a review [J]. Mathematics, 2022, 10 (8): 1206.

[4] Tavana M, Soltanifar M, Santos-Arteaga F J. Analytical hierarchy process: Revolution and evolution [J]. Annals of operations research, 2023, 326 (2): 879-907.

[5] Leal J E. AHP-express: A simplified version of the analytical hierarchy process method [J]. MethodsX, 2020, 7: 100748.

[6] Ghosh M. Bayesian methods for finite population sampling [M]. Routledge, 2021.

[7] Lye A, Cicirello A, Patelli E. Sampling methods for solving Bayesian model updating problems: A tutorial [J]. Mechanical Systems and Signal Processing, 2021, 159: 107760.

[8] Van de Schoot R, Depaoli S, King R, et al. Bayesian statistics and modelling [J]. Nature Reviews Methods Primers, 2021, 1 (1): 1.

[9] Gagné C, Sioud A, Gravel M, et al. Multi-objective optimization [J]. Heuristics for Optimization and Learning, 2020, 906: 183.

[10] Wang, F., Guo, L., & Zhao, W. A Comprehensive Study on Micro-Defect Detection in PCB Using Convolutional Neural Networks [J]. Sensors, 2021, 21 (8): 2896.

[11] Chen, J., Liu, Z., & Wu, T. A Dynamic Block Strategy for Micro-Defect Localization in PCB Inspection Based on CNN [J]. Journal of Electronic Testing, 2023, 39 (3): 312-325.