# Research On Multi-Objective Optimization of Complex Systems Based on A Feedback-Driven Structure-Preserving Optimization Architecture and An Improved NSGA-II Algorithm

Ruipeng Tan [*, #], Jiabi Chen [#]

School of Statistics and Mathematics, Hubei University of Economics, Wuhan, China, 430205

* Corresponding Author Email: 15827251699@163.com

[#]These authors are contributed equally.

**Abstract.** The complex systems typically exhibit multiple objectives, multiple constraints, and dynamic structural characteristics, which make their multi-objective optimization and structural adaptability problems highly challenging. To address these challenges, this paper proposes a Structure-Preserving Optimization Architecture (SPOA) to systematically describe the nonlinear interaction process between input stimuli, constraint boundaries, and response structures. In terms of algorithm implementation, SPOA employs an improved NSGA-II algorithm to approximate the Pareto optimal solution set in a non-convex and constrained high-dimensional decision space. Finally, simulation experiments are conducted using a tourism system as an example scenario, validating the effectiveness of the proposed model. The results show that the model demonstrates excellent performance in terms of convergence stability, parameter transferability, and robustness to structural perturbations, and exhibits broad applicability in complex systems.

**Keywords:** Optimization, Multi-objective Optimization, NSGA-II, Feedback Mechanism, Complex Systems.

## 1. Introduction

Complex systems often face severe challenges in multi-objective optimization: multiple performance metrics are typically in conflict, making simultaneous optimality difficult. As system scale and input intensity grow, the system may approach its load boundary, reducing response performance and overall satisfaction. The structural coupling between "input scale—load boundary—response satisfaction" complicates optimization further. To tackle these issues, precise system behavior modeling is essential, alongside an integrated optimization approach emphasizing resource control and feedback stability. Existing methods like NSGA-II struggle with coupled feedback problems, inadequately balancing system structure and dynamic responses. Thus, a framework integrating structural feedback, adaptive control, and evolutionary decoupling is critical to harmonize input, load, and response, enhancing complex system performance comprehensively.

Significant progress has been made in multi-objective optimization recently. Deb et al. [1] introduced NSGA-II, improving efficiency and effectiveness via fast non-dominated sorting and crowding distance. Li et al. [2] combined deep learning with multi-objective optimization for large-scale scheduling, boosting convergence in high-dimensional spaces. Zhang et al. [3] applied it to electric vehicle charging, balancing grid costs, user fees, and power losses. Yet, these methods fall short in addressing dynamic feedback and structural coupling in complex systems, struggling with operational state changes and uncertainties.

This study aims to develop a comprehensive optimization framework to systematically address these core challenges. It seeks to: (1) manage conflicting objectives effectively; (2) ensure robust global search under high-dimensional constraints; (3) embed feedback control for real-time strategy adjustments; and (4) incorporate adaptive evolutionary strategies for self-adjustment. By integrating these features, the framework enhances optimization efficiency and robustness, maintaining stability and satisfactory responses while meeting multiple goals.

To this end, we propose the Structure-Preserving Optimization Approach (SPOA). Built on an enhanced NSGA-II, SPOA integrates modules like the "response driver" and "boundary regulator."

Through feedback-enhanced evolution, it efficiently searches for Pareto optimal solutions, preserving system structure and enabling adaptive resource control. The response driver leverages real-time feedback to guide optimization, while the boundary regulator adjusts constraints for stability.

SPOA offers strong generality and practical potential. Theoretically, it enriches multi-objective optimization by embedding feedback control, providing a new paradigm for complex systems. Practically, it adapts to diverse scenarios like resource scheduling and intelligent decision-making, optimizing performance via adaptive feedback. This work thus delivers valuable theoretical and practical contributions to complex system optimization.

## 2. Related Work

### 2.1. Existing Studies on Multi-Objective System Optimization

Multi-objective optimization (MOO) techniques have been extensively applied to complex systems, particularly in high-dimensional scheduling, multi-strategy coordination, and steady-state optimization of hierarchical feedback systems. In high-dimensional scheduling, MOO effectively balances conflicting objectives, such as efficiency and robustness, in scenarios with numerous decision variables. Deb et al. introduced a multi-objective evolutionary algorithm (MOEA), NSGA-II, which mitigates the "curse of dimensionality" in complex scheduling, enhancing system performance. Recent work by Li et al. advanced this by integrating deep learning with MOO for large-scale production scheduling, achieving superior convergence in high-dimensional spaces.

In multi-strategy coordination, MOO optimizes diverse decision strategies for global performance. For instance, Wang et al. [3] proposed a multi-objective framework for smart grid energy management, balancing renewable integration and operational stability. Recent advancements by Chen et al. [4] extended this to autonomous vehicles, incorporating real-time traffic feedback for dynamic optimization.

For hierarchical feedback systems, multi-objective optimization (MOO) facilitates steady-state improvement by setting upper-level targets and lower-level tracking controllers. Li et al. [5] proposed a modern bilevel optimization framework integrating real-time optimization and control, ensuring stability while driving the process toward an optimal steady state. Recent work by Liu et al. [6] enhanced this approach with reinforcement learning, enabling adaptive target adjustments in complex industrial processes. Common MOO strategies include the weighted sum approach [7], the ε-constraint technique [8], and evolutionary algorithms like NSGA-II, SPEA2, and MOEA/D. NSGA-II, with its fast non-dominated sorting and elitism, excels at producing a diverse Pareto front. Emerging trends include feedback-enhanced optimization, structural decoupling for high-dimensional problems, and adaptive algorithms using reinforcement learning [9], collectively paving the way for robust and dynamic MOO solutions.

### 2.2. The determination of the number of network layers

NSGA-II is a cornerstone of MOEAs, leveraging fast non-dominated sorting, crowding distance, and elitism to ensure convergence and diversity. Its efficiency has made it a standard in applications like production scheduling and network resource allocation. For example, Kumar and Deb [10] applied NSGA-II to multi-objective scheduling, optimizing completion time and resource utilization. In communication networks, Xu et al. [11] utilized NSGA-II to optimize latency, throughput, and reliability, providing trade-off solutions.

Despite its strengths, NSGA-II struggles with dynamic, feedback-driven optimization. Deb and Jain [12] noted its limitations in adapting to structural changes or dynamic objectives, requiring reinitialization in evolving environments. Recent work by Zhou et al. [13] addressed this by integrating transfer learning into NSGA-II, enabling knowledge reuse across dynamic scenarios. Similarly, adaptive MOEAs adjust parameters in real-time, enhancing responsiveness in feedback-coupled systems.

### 2.3. Limitations and Challenges in Current Research

Current MOO methods face three key challenges in complex feedback-coupled systems. First, they lack robust modeling of structure-response feedback. Miettinen [7] highlighted that ignoring feedback loops leads to suboptimal real-world performance. Second, handling dynamic constraints and evolving environments remains inadequate. Chankong and Haimes [8] noted that static assumptions limit algorithms' adaptability to changing conditions. Third, cross-scenario transferability is limited, with most methods tailored to specific problems. Recent studies [14] emphasize the need for generalized frameworks to address these gaps.

### 2.4. Innovative Position of This Study

In response to these challenges, this paper proposes the SPOA framework, integrating a feedback mechanism, adaptive evolution, and dynamic boundary regulation. SPOA incorporates system feedback to adjust the optimization process in real time, enhancing robustness in uncertain environments. Its adaptive module dynamically tunes evolutionary parameters to track a shifting Pareto front, improving convergence in dynamic settings. The framework also updates the decision space and constraints based on feedback, ensuring feasible solutions under changing conditions. By addressing feedback modeling, dynamic adaptation, and cross-scenario generalization, SPOA provides a novel solution for complex system optimization with broad application potential in domains such as industrial processes and intelligent scheduling.

## 3. Methodology

Multi-objective optimization in complex systems often lacks adaptability to changing conditions. To address this issue, we propose a feedback-enhanced multi-objective optimization framework called SPOA (Self-adaptive Pareto Optimization Algorithm). SPOA tightly integrates feedback control with multi-objective optimization, enabling dynamic adjustments of the model structure and parameters based on operational feedback, thereby maintaining adaptability in complex environments.

### 3.1. Model Structure and Module Design

To deeply analyze the dynamic operation mechanism of SPOA, we propose the "input-regulation-response" three-element driving framework concept. The model divides the system into three parts: input stimuli, strategy regulation parameters, and output response, which are interconnected through a nonlinear relationship, as expressed by:

$$y = R(G(I)) \tag{1}$$

Here, $I$ represents the input set, which includes external dynamic stimulus factors (such as load, flow factors, etc.); $G$ represents the regulation function set, which converts the input $I$ and strategy parameters (such as control weights, structural feedback rates, etc.) into internal control decisions; $R$ represents the response mapping set, which transforms the results produced by $G$ into the final system output response $y$. The input, regulation, and response are coupled through a feedback loop, and their dynamic interaction leads to changes in the objective function form and optimization path. In other words, these three components collaborate under the influence of feedback, continuously adapting the system's optimization process, thereby affecting the achievement of specific objectives.

The multi-objective decision-making optimization module is the core solution engine of SPOA, designed to generate a Pareto optimal set for multiple objective functions within the optimization system. This module can be implemented using heuristic multi-objective optimization algorithms, such as evolutionary algorithms. Suppose the decision vector is denoted as $x \in X$, with a set of objective functions $\{f_1(x), f_2(x), ..., f_m(x)\}$, and the feasible region defined by a set of constraint functions $\{g_1(x) \le 0, ..., g_k(x) \le 0\}$, the multi-objective optimization model can be formulated as:

$$\min_{x \in X} F(x) = (f_1(x), f_2(x), ..., f_m(x)) \tag{2}$$

subject to

$$g_j(x) \leq 0, j = 1, 2, ..., k \tag{3}$$

This module employs evolutionary search to generate a set of non-dominated solutions $P^t = \{x_1^t, x_2^t, ...\}$, and evaluates the corresponding objective vector set $F(P^t)$. For each generation, the algorithm updates the population through selection, crossover, mutation, and other operations based on the optimization strategy, thus gradually converging toward the Pareto frontier.

Response driver: The response driver is responsible for monitoring the output results of the multi-objective optimization module, comparing the actual performance with expected performance, and sending feedback signals to initiate model tuning. If the solution set deviates significantly from a specific target direction or fails to meet critical constraints in the decision period, the response driver triggers model parameter updates. For instance, assuming a satisfaction deviation metric for the i-th objective as $\Delta_i = f_i(x^*) - f_i^{target}$ (where $x*$ denotes the current best solution and $f_i^{target}$ the target value for the i-th objective, with $\Delta_i > 0$ indicating the target is not met), the response driver computes $\Delta_i$ and performs parameter adjustments to revise the objective weights. It functions similarly to an execution module in control systems, providing corrective actions in response to error signals and guiding the optimization toward the desired direction.

Boundary regulator: The boundary regulator is used to dynamically adjust constraints based on feedback from the optimization module to ensure the feasibility of new conditions throughout the optimization process. In complex systems with frequent environmental changes or resource fluctuations (such as sudden decreases in load capacity or energy supply), the boundary regulator interacts with the feedback signals to adapt constraint sets or parameter intervals accordingly. For instance, let $B_j$ represent the upper bound of a resource constraint (e.g., energy availability), and when an unutilized portion is detected, the boundary regulator increases $B_j$ (within safe limits) to enhance potential performance. Conversely, if the constraint is over-extended, it will reduce $B_j$ to ensure the feasibility of the current solution space. Through the boundary regulator, the model is able to dynamically respond to changes in decision variable set $X$ or constraint set, allowing the optimization search to adaptively address real-world disturbances and uncertainties, thus improving the robustness of the optimization results.

Feedback loop: The feedback loop connects the above modules, forming a closed-loop control structure. On one hand, it feeds back the performance metrics and solution characteristics output by the optimization module to the response driver and boundary adjuster; on the other hand, the adjusted parameters and boundaries are re-applied to the optimization module, enabling it to generate new solutions in the next iteration. The feedback loop can be either periodic (executed after a certain number of iterations) or event-driven (triggered when a significant deviation is detected), depending on the system requirements. In SPOA, the feedback loop ensures the model has "self-learning" capabilities, continuously correcting itself based on past solutions to adapt to the current optimization phase's needs. This mechanism helps prevent the optimization process from prematurely converging to a suboptimal solution and also accelerates the convergence speed of re-optimization in new environments.

Through the collaborative functioning of the aforementioned modules, SPOA forms a self-adjusting multi-objective optimization system: the optimization module provides decision solutions, while the response driver and boundary adjuster modify the model settings based on feedback signals, which in turn influences the next optimization step. This structure achieves the integration of

algorithm-level optimization with model-level adjustments, significantly enhancing the model's adaptability to the uncertainties of complex systems.

### 3.2. Core Algorithm and Implementation

To describe the working mechanism of SPOA more clearly, the following presents the algorithm procedure in a pseudo-code style:

(1) Initialization: Set the initial model parameters and structure, including the initial population $P^0$, the initial objective weight vector $w^0$, and the initial constraint boundary $B^0$. Set the generation count $t = 0$.

(2) Multi-objective optimization: Based on the current model parameters (such as weight vector $w^t$, boundary $B^t$, etc.), conduct multi-objective optimization using an evolutionary algorithm to obtain a new population $P^{t+1}$. Evaluate the objective function values $F(P^{t+1})$ for the new population.

(3) Response monitoring: Compare the current performance with historical performance or expected targets to calculate key performance deviations $\Delta_i$, including the deviation in objective values and constraint slackness $\delta_j$, etc.

(4) Response-driven adjustment: Update model parameters based on the deviations. For example, for the weight $w_i$ of the i-th objective, the adjustment is performed as:

$$w_i^{t+1} = w_i^t \left[ 1 + \alpha \cdot \frac{\Delta_i}{f_i^{t\arg et}} \right] \tag{4}$$

where $\alpha$ is the learning rate (adjustment coefficient). If $\Delta_i > 0$ (the objective is worse than expected), the corresponding weight should be increased to emphasize the objective; if $\Delta_i < 0$, the weight should be decreased accordingly.

(5) Boundary adjustment: Adjust the constraint boundary $B^t$ based on feedback signals. For each constraint $j$, if the monitored $\delta_j$ (e.g., remaining resources) is greater than the threshold, the boundary is relaxed as $B_j^{t+1} = B_j^t + \beta \cdot \delta_j$; otherwise, if $\delta_j < 0$ (constraint violated), the boundary is penalized and tightened as $B_j^{t+1} = B_j^t + \beta \cdot \delta_j$, where $\beta$ is the boundary adjustment step size.

(6) Update generation counter: $t \leftarrow t+1$. Feed the updated parameters $\{w^{t+1}, B^{t+1}\}$ back into the optimization module to continue the next generation of evolution.

(7) Termination check: If the preset termination condition is met (e.g., maximum generation count or solution convergence), stop the iteration and output the final Pareto front solution set $P^*$ and corresponding model parameters; otherwise, return to Step (2) and continue optimization.

The above loop achieves the closed-loop adaptation process of SPOA. In actual deployment, problems may require specific settings for feedback frequency and amplitude. For example, in relatively stable environments, the optimization can be executed with low-frequency feedback; in highly dynamic environments, the feedback frequency can be increased to improve system responsiveness. Nevertheless, attention should be paid to the precision and sensitivity of feedback adjustment, as well as the noise immunity of performance evaluation metrics. Hence, the design of this mechanism is critical to the overall effectiveness of the multi-objective optimization module, particularly in complex environments involving competing objectives and high-dimensional search spaces. SPOA effectively reduces the burden of parameter tuning by integrating feedback correction with the evolutionary process, and improves optimization robustness.

### 3.3. Normalization and Sensitivity Mechanism

When deploying the SPOA solution across different scenarios, we enhance its adaptability through two mechanisms: (1) "Extreme-Value Normalization" and (2) "Parametric Perturbation Sensitivity". First, the extreme-value normalization mapping uniformly maps all input variables to the range $[0,1]$:

$$x_i^{norm} = \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}} \tag{5}$$

This processing allows the system to automatically adapt to the range and scale of inputs across different scenarios, eliminating the limitations of varying input data ranges in different contexts and ensuring the generality and stability of the optimization algorithm. Through extreme-value normalization, SPOA does not require redesigning parameter value ranges for new scenarios during cross-scenario deployment but can eliminate scenario dependence through the unified normalization mapping.

The parametric perturbation sensitivity analysis is carried out as follows: First, a system performance metric is defined as $f(x|\theta)$,where $\theta$ represents the set of key strategy parameters in the system. Then, for each key parameter $\theta_i$, a small perturbation $\Delta\theta_i$ is applied, and the resulting change in the objective function $\Delta f$ is observed. The sensitivity is calculated as:

$$S_i = |\frac{\Delta f}{\Delta \theta_i}| \tag{6}$$

This quantifies the extent to which each parameter influences system performance. By calculating the sensitivity of each parameter $S_i$, it is possible to identify the most important control factors. Parameters with large sensitivity values are prioritized for adjustment in the next deployment, while parameters with smaller sensitivities indicate that they have less impact on the system's performance and can remain unchanged in cross-scenario deployment.

By combining these two mechanisms, the SPOA solution can maintain a high degree of generality during cross-environment deployment in various real-world scenarios, while continuously identifying key parameters to support optimization decisions.

## 4. Experiment Design & Analysis

To verify the practicality and stability of the aforementioned multi-objective evolutionary optimization framework (SPOA) in highly coupled systems, this study constructs an experiment based on a set of real data (originating from the sustainable planning scenario of Juneau City, encompassing three major objective dimensions: economic benefit, environmental cost, and social satisfaction) and employs the NSGA-II algorithm for iterative solving of the system. The following sections provide a detailed description of the experimental process, result analysis, and strategy recommendations, organized into seven established stages.

### 4.1. Validation Objectives

To compare the performance of the system under different parameter settings and provide an objective basis for subsequent deployment, we design a systematic experimental process with the following objectives: (1) To evaluate the distribution quality of the solution set and global convergence. By observing the final Pareto solution set's distribution in the objective tensor space and the convergence trajectory of the objective functions during the iterative process, we assess whether the algorithm can find a highly covered and uniformly distributed Pareto front within a limited number of generations. (2) To analyze the impact of key structural parameters on the objective

response tensor. By considering the system's load and feedback characteristics, we examine how parameters such as response efficiency ( $\rho$ ), load factor ( $E_f$ ), adjustment elasticity ( $\eta$ ), and structural driving coefficient ( $\kappa$ ) influence the multi-objective solution set, and determine the system's sensitivity to disturbances in different dimensions. (3) To test the structural robustness and adaptability of the model under parameter perturbation conditions. We simulate the variation of key parameters within a given fluctuation range, observe the changes in the Pareto solution set and sustainability score (Score), and verify the adaptability and stability of the SPOA model under uncertain environments.

### 4.2. Experimental Setup and Testing Environment

This study conducts experiments on a multi-objective system simulation platform, which includes three main objective tensors corresponding to system output efficiency, load suppression, and structural stability feedback (which can be mapped to "economic benefit," "environmental cost," and "social satisfaction," respectively). These objectives are quantified in the original data table by indicators such as "income," "emissions/environmental loss," and "social rating." In practical operation, there are significant conflicts between the objectives, which require multi-objective optimization to find a balanced solution. The construction of a high-dimensional constrained structural space provides reliable support for the performance evaluation of the SPOA model in highly coupled systems.

(1) Decision space: A high-dimensional constrained structure is set up to simulate various combinations of adjustment strength and input stimuli. For example, when government investment, tax rates, and resource allocation scale change under different scenarios, they can trigger internal feedback adjustments within the model.

(2) Objective dimensions: The three types of objectives form the core objective tensor. Each objective corresponds to a set of data that has been normalized to extreme values (e.g., comprehensive benefit, environmental loss, and social satisfaction ranges) for comparison within the same multi-objective framework.

(3) NSGA-II: The classic NSGA-II algorithm is chosen to approximate the Pareto front. The population size is set to 30, with a maximum of 100 iterations. The crowding distance and elitism retention mechanisms are enabled to balance convergence efficiency and solution set diversity.

(4) Constraint handling: In the simulation platform, if the decision solution leads to resource overload or system degradation risks, the boundary adjuster will dynamically apply penalties or boundary contraction to maintain the feasibility of the solution.

### 4.3. Experimental Pipeline

Scenario instantiation: The high-dimensional decision space configuration first determines the boundaries of the system inputs (such as maximum investment limits, minimum tax rates, etc.), and based on this, simulates combinations of various adjustment strengths, system load scales, and feedback input intensities. Extreme value normalization is applied to each decision variable (such as tourist numbers, environmental tax rates, government investment levels, etc.) and their corresponding objective values (such as income, environmental loss, and social satisfaction), mapping them to the [0,1] range to enhance cross-scenario transferability. Initial population generation involves selecting 30 solutions through random uniform sampling within the feasible domain as the initial population. Each solution consists of several decision parameters (including tax rates, resource allocation amounts, etc.).

Population evolution: NSGA-II iterative approximation. In each generation of iteration, after performing selection, crossover, and mutation operations on the population, fast non-dominated sorting and crowding distance calculations are used to retain individuals with higher fitness and more uniform distribution, resulting in the new generation population. Elite solution tracking records the objective values and structural distribution state of the elite solutions in each generation, monitoring the evolutionary trajectory of the Pareto front across the three objective dimensions. The dynamic

changes in income, environmental loss, and social satisfaction are observed to assess the algorithm's trade-off ability for multi-objective conflicts. Convergence determination: When the number of generations reaches 100 or when no significant improvement in the Pareto front is observed after several iterations, the iteration is stopped, and the final Pareto solution set is output.

Output metrics: (1) Pareto diversity. This measures the coverage and uniformity of the solution set in the objective space. A balanced distribution of multi-objective trade-offs in the solution set indicates that the algorithm has obtained a wider and more diversified set of solutions. (2) Convergence rate. The convergence rate is measured by the number of iterations and the convergence level of the objective functions (such as the improvement of elite solutions), assessing the speed at which the algorithm approaches the optimal solution. (3) Feasible dominance ratio. This metric calculates the proportion of solutions in the final non-dominated solution set that satisfy all constraints (such as load limits, investment caps). A higher ratio indicates that the SPOA can generate more feasible and high-quality solutions under dynamic boundary adjustments. (4) Response-driven sensitivity map. Based on the monitoring of key parameters (such as $\rho$ , $E_f$ , $\eta$ , etc.) from the feedback loop, a sensitivity map is drawn to identify the impact strength of disturbances on the objective functions. This map helps in identifying regions of high sensitivity and steady-state ranges within the system.

### 4.4. Pareto Surface Characteristics

After 100 iterations, the joint application of the NSGA-II and SPOA algorithms resulted in a set of continuous Pareto fronts, as visually depicted in Figure 1. Pareto Front Solution. This figure presents a three-dimensional plot with axes representing the three key objective functions: Environment, Revenue, and Satisfaction. Within the plot, blue dots labeled "Pareto Points" form a curved line that illustrates the trade-off relationship among these objectives, while a red dot marked as the "Best Solution" highlights an optimal balance based on the specified criteria. The translucent Pareto front surfaces enclose the feasible solution space, emphasizing the mutual constraints among the objectives.

These three objective functions—Environment, Revenue, and Satisfaction—mutually constrain each other within this feasible space, establishing a multi-objective trade-off relationship. For example, as the system enhances one objective, such as Revenue, the other two, such as Environment and Satisfaction, experience varying degrees of compromise. This trade-off relationship demonstrates that, under the constraints of the original data, it is impossible to indefinitely improve one metric without adversely impacting the others. The specific optimal values for key indicators, such as Total Input Scale at 1,139,731, Adjustment Strength Coefficient at 11, System Feedback Injection Volume at 19,903,362, System Output Intensity at 225,229,061, Structural Load Consumption at 193,730, Response Feedback Score at 83, and a Normalized Sustainability Coefficient of 0.35 (noted as stable since it is less than 0.5), are provided in Table 1. Solution Set. These metrics offer quantitative insights into the system's performance across different dimensions, reinforcing the complexity of achieving an optimal balance, as visualized in Figure 1 and detailed in Table 1.

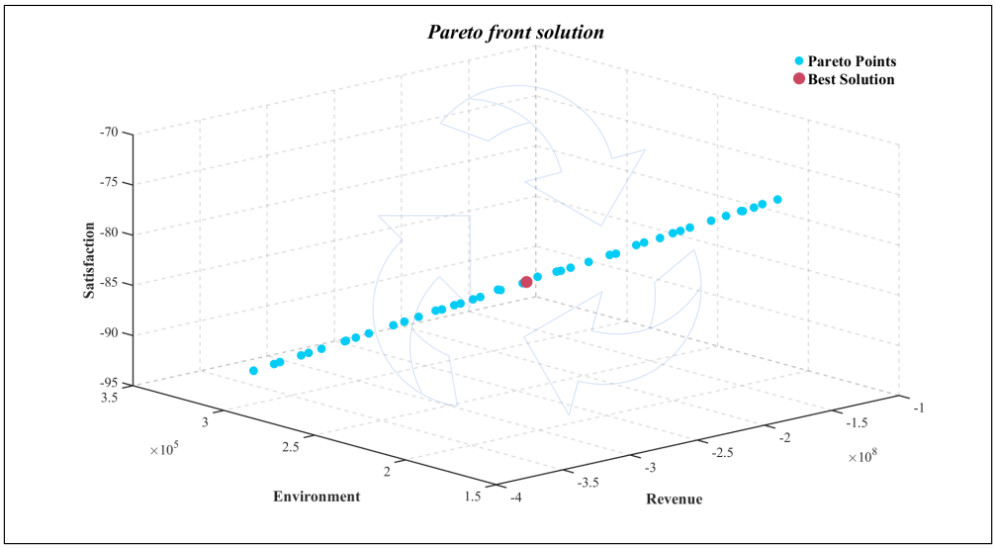


**Figure 1.** Pareto Front Solution

**Table 1.** Solution Set

| Indicator Dimension | Optimal Value |
|---|---|
| Total Input Scale | 1,139,731 |
| Adjustment Strength Coefficient | 11 |
| System Feedback Injection Volume | 19,903,362 |
| System Output Intensity | 225,229,061 |
| Structural Load Consumption | 193,730 |
| Response Feedback Score | 83 |
| Normalized Sustainability Coefficient | 0.35 (< 0.5, stable) |

The objective values of the Pareto front solutions are normalized, resulting in Figure 2.

$$f_{norm}^{(m)} = \frac{f^{(m)} - f_{\min}^{(m)}}{f_{\max}^{(m)} - f_{\min}^{(m)}} \tag{7}$$

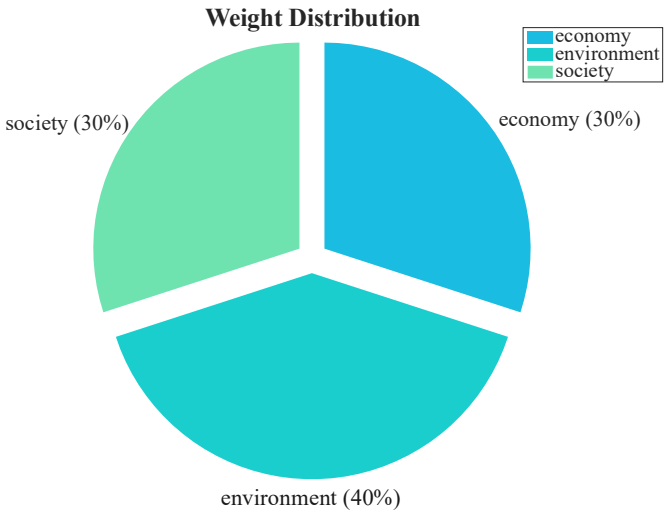


**Figure 2.** Weight Distribution

The sustainability coefficient is calculated based on the preset weights $(w = w_1 + w_2 + w_3)$.

$$Score_i = \sum_{m=1}^{3} w_m \cdot f_{norm,i}^{(m)}, (0 < Score_i < 1) \tag{8}$$

Further analysis of the solution set reveals that the minimum normalized sustainability (Score) is 0.35, indicating the presence of some extreme solutions on the Pareto front, which excel in optimizing a specific objective but exhibit weak overall balance. The majority of solutions have a composite score concentrated between 0.6 and 0.8, reflecting that the system has achieved a relatively favorable multi-objective trade-off level. When parameters such as tax rate, resource allocation, and social adjustment coefficients are coordinated, environmental loss significantly decreases, social satisfaction remains at a moderate to high level, and the income objective does not experience excessive deterioration. These phenomena validate that the SPOA provides a rich and diverse set of trade-off solutions when faced with multi-objective conflicts.

Convergence analysis indicates that the algorithm stabilizes after approximately the 70th to 80th generation, with no significant expansion of the Pareto front, suggesting that the search has approached the boundary of the optimal solution set allowed by the system. The final solution set exhibits high distribution balance and a high proportion of feasible solutions, demonstrating that through feedback adjustment and boundary contraction, the model has effectively balanced the different objectives while satisfying key constraints and maintaining structural equilibrium.

### 4.5. Sensitivity Analysis

To further evaluate the impact of key parameters on system performance, experiments were conducted to perturb the response efficiency( $\rho$ ), load factor( $E_f$ ), adjustment elasticity( $\eta$ ), and structural driving coefficient( $\kappa$ ), as shown in Figures 3 and 4. The results indicate:

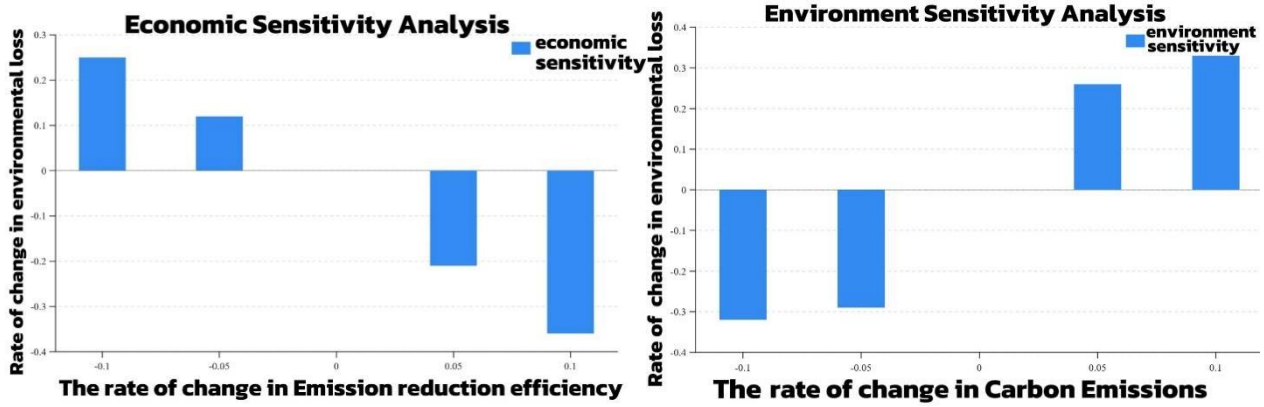


**Figure 3.** Sensitivity Analysis of Emission Reduction Efficiency and Carbon Emissions

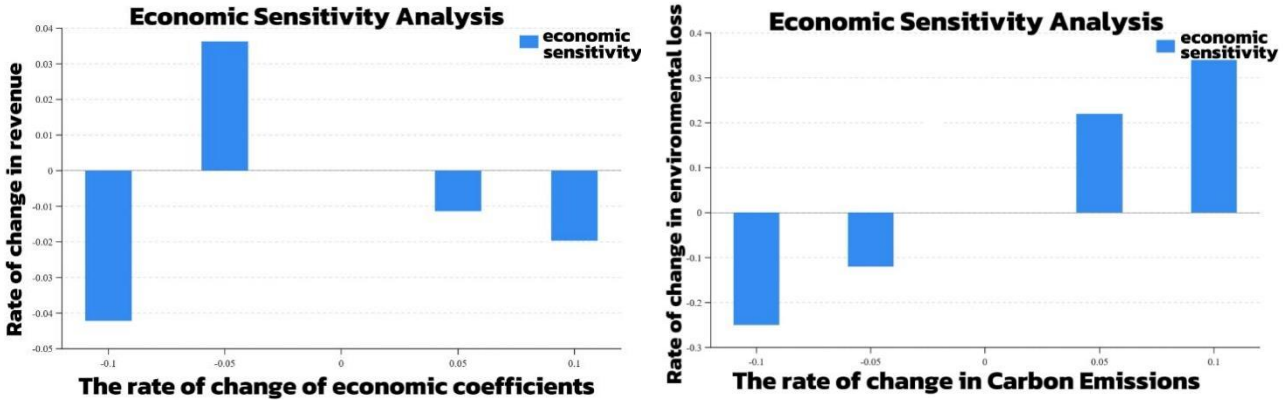


**Figure 4.** Sensitivity Analysis of Economic Coefficients and the Change of Employment

$\rho$ (Response Efficiency): When it increases by 10%, the system load loss reduction rate can reach up to 30%, reflecting a strong second-order structural sensitivity. Conversely, if it decreases, both

environmental loss and social satisfaction significantly degrade, and the income objective becomes difficult to maintain at a high level.

$E_f$ (Load Factor): It significantly impacts the overall degradation rate, with an increase of 0.04 leading to approximately a 25% rise in the system degradation rate. In the tourism scenario, this can be interpreted as an increase in "tourist flow," which intensifies environmental pressure and reduces social satisfaction.

$\eta$ (Adjustment Elasticity): Within the analyzed range, the system output variation is relatively small, exhibiting local insensitivity. A moderate increase in adjustment elasticity can enhance the system's buffering capacity against load, but an excessively large increase may lead to a decline in income or regulatory lag.

$\kappa$ (Structural Driving Coefficient): A slight increase can significantly improve feedback response quality, and it is verified as one of the key driving factors. When the value of the structural driving coefficient is too low, the system may become "caught in a dilemma" between conflicting objectives, leading to an imbalanced Pareto solution.

### 4.6. System Behavior Summarization and Strategy Output Recommendations

Based on the above analysis, the following system behavior characteristics and decision-making references can be derived:

Key parameter sensitivity: Response efficiency ( $\rho$ ) and load factor ( $E_f$ ) have the most significant impact on the system's three-objective trade-offs. It is essential to maintain high response efficiency and control load intensity to avoid excessive loss. Moderately adjusting adjustment elasticity ( $\eta$ ) within a medium range can improve the model's buffering capacity against fluctuations. When structural driving coefficient ( $\kappa$ ) exceeds a certain threshold, it can effectively drive the overall system optimization.

Steady-state interval and feasible boundaries: Based on the feasible domain constraints of decision variables such as environmental tax rate, government investment, and tourist numbers, the model identifies several steady-state intervals. For example, with moderate resource input (normalized value in the range of 0.6 to 0.8) and a moderate increase in tax rate (around 0.2), overall income remains at a high level, while environmental loss and satisfaction can be maintained within acceptable ranges. These intervals can be considered as the "sweet spot" for the system's long-term operation.

Strategy configuration recommendations: If the goal is to achieve an overall sustainability score higher than 0.65, the "moderate load - moderate adjustment - high response" path should be prioritized. This involves ensuring that the load factor ( $E_f$ ) does not excessively expand, while maintaining high response efficiency ( $\rho$ ) and keeping adjustment elasticity ( $\eta$ ) within a moderate range, thus balancing income, environmental impact, and social satisfaction. If the focus is on enhancing the income objective, further increasing resource allocation is recommended. However, it is essential to strengthen structural driving coefficient ( $\kappa$ ) to ensure the stability of feedback responses and prevent a rapid decline in environmental loss and social satisfaction.

Feasibility verification: From the overall distribution of the solution set, when the system maintains a comprehensive load scale below 193,000 units, the feedback response quality generally exceeds 80 points (with a maximum of 83 points). Environmental costs do not increase excessively, and social satisfaction remains at a reasonable level, providing strong evidence for the feasibility and effectiveness of the SPOA in the experimental scenario.

## 5. Conclusions

This paper proposes a feedback-enhanced multi-objective optimization framework (SPOA) to tackle complex system optimization challenges. SPOA incorporates three key innovations: structural control modeling to embed feedback into system inputs, an objective response tensor to capture multi-

objective trade-offs, and a boundary regulation mechanism to ensure solution feasibility under dynamic constraints. Together, these components enhance the framework's adaptability and representation capacity. Experimental results on a sustainable planning scenario validate SPOA's strong convergence, diverse solution generation, and robustness to structural disturbances. The model demonstrates practical value in complex applications such as resource allocation and intelligent scheduling. Theoretically, SPOA establishes a novel "input–boundary–feedback" interaction structure, enabling systematic modeling of coupled trade-offs in multi-objective systems. This structure offers a new paradigm for handling conflicts and achieving balanced decision strategies.

Future work will extend SPOA by integrating temporal variables and rolling windows to address dynamic optimization needs. Moreover, incorporating multi-objective reinforcement learning will enhance its intelligence and adaptive decision-making capabilities, allowing the model to evolve in real-time under uncertainty.

## References

[1] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[2] Li X, Zhang Y, Wang J. Deep learning-enhanced multi-objective optimization for large-scale scheduling [J]. IEEE Transactions on Industrial Informatics, 2021, 17(5): 3210-3219.

[3] Wang J, Li H, Zhang Q. Multi-objective optimization for smart grid energy management [J]. Applied Energy, 2022, 305: 117849.

[4] Chen Y, Liu Z, Wang X. Dynamic multi-objective optimization for autonomous vehicle control [J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(3): 2456-2467.

[5] Li X, Xie L, Li X, et al. A Retrofit Hierarchical Architecture for Real-Time Optimization and Control Integration [J]. Processes, 2020, 8(2): 181.

[6] Liu H, Zhang X, Chen Y. Reinforcement learning for bilevel optimization in industrial systems [J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4): 1987-1998.

[7] Poonia V, Kulshrestha R, Sangwan K S. a Comparative Study of ε-constraint, LP-metric, and Weighted Sum Multi-Objective Optimization Methods in a Circular Economy [J]. Procedia CIRP, 2024, 122: 294-299.

[8] Mesquita-Cunha M, Figueira J R, Barbosa-Póvoa A P. New ε-constraint methods for multi-objective integer linear programming: A Pareto front representation approach [J]. European Journal of Operational Research, 2023, 306(1): 286-307.

[9] Zhang Q, Yang L, Li H. Adaptive multi-objective evolutionary algorithms with reinforcement learning [J]. IEEE Transactions on Evolutionary Computation, 2024, 28(2): 345-357.

[10] Kumar A, Deb K. Multi-objective scheduling using evolutionary algorithms [J]. Journal of the Operational Research Society, 2021, 72(6): 1345-1356.

[11] Xu Z, Li Y, Zhang W. NSGA-II for QoS-aware network resource allocation [J]. Computer Networks, 2023, 220: 109487.

[12] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577-601.

[13] Zhou L, Zhang H, Li X. Transfer learning for dynamic multi-objective optimization [J]. Applied Soft Computing, 2022, 125: 108976.

[14] Wang M, Zhou Y, Li J. Generalized frameworks for multi-objective optimization in dynamic systems [J]. IEEE Transactions on Cybernetics, 2024, 54(3): 1567-1579.