Journal of Engineering, Project, and Production Management
2026, 16(1), 2025-155

Improved Multi-Objective Particle Swarm Optimization for Sustainable Building Design

Song Du
Deputy Director, Infrastructure Construction Department, Jiangsu Maritime Institute, Nanjing, 211170, China, E-mail: dnfdusong@outlook.com

Project Management
Received August 10, 2025; revised September 25, 2025; accepted September 26, 2025
Available online December 24, 2025

Abstract: Traditional multi-objective optimization methods suffer from limitations such as sluggish convergence and high scenario sensitivity when it comes to multi-objective architectural design decision optimization for sustainable buildings. This study constructs a multi-objective architectural design decision optimization model that simultaneously considers comfort level and building energy consumption. The model is solved using a modified backbone multi-objective particle swarm optimization algorithm that optimizes the search strategy and introduces an adaptive perturbation mechanism. Results show the improved algorithm achieved higher hypervolume values than the rest of the mainstream algorithms in both building scenarios. Performance remained stable with minimal fluctuation between scenarios. In the single-room office scenario, the average hypervolume value of the improved algorithm was as high as 29,963. This substantially exceeded the non-dominated sorting genetic algorithm II (NSGA-II), which achieved 19,246. For the three residential scenarios, the improved algorithm reached an average hypervolume of 42,639, compared to 14,628 for standard multi-objective particle swarm optimization. Across scenarios, the improved algorithm's average run times (1.38h and 3.38h, respectively) were lower than all other algorithms. In addition, the algorithm's Pareto frontier solutions were concentrated in the low-energy, high-comfort region. In conclusion, the improved algorithm effectively achieves dual-objective decision optimization, balancing user comfort with building energy efficiency. Novelty lies in integrating a backbone guidance mechanism with an adaptive perturbation strategy. This addresses parameter redundancy and premature convergence in traditional multi-objective particle swarm optimization. The approach practically enables rapid generation of energy-efficient designs while maintaining high comfort levels. This provides architects with quantitative support for sustainable design across various building types, including offices and residences.

Keywords: Sustainable building, PSO, multi-objective, decision optimization, adaptive perturbation.

DOI 10.32738/JEPPM-2025-155

1. Introduction

The building sector represents a significant source of Energy Consumption (EC) and carbon emissions. Amplified by global climate change and energy crisis, sustainable development in this sector has become crucial for achieving the "double carbon" goal (Zhong et al., 2024). Building operating EC constitutes over 30% of global EC, while associated carbon emissions account for over 28%, according to the International Energy Agency figures. Moreover, the acceleration of urbanization continues to drive these figures upward. In the context of rising global building EC and tightening carbon constraints, sustainable building design is urgently needed to reduce EC and environmental impacts while taking into account multiple objectives such as Comfort Level (CL) and construction costs (Ramzanpoor et al., 2022; Mangalampalli et al., 2022). Among traditional decision optimization methods, mathematical planning requires preset objective weights, limiting its effectiveness in dynamic and complex scenarios. The hierarchical analysis method relies on subjective assignment, introducing potential decision bias. Although Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) enable global search, their performance is sensitive to the values of Inertia Weights (IWs), learning factors and other control parameters. Their manual tuning cost is extremely high. Meanwhile, the necessity of continually using expensive EC simulation software leads to time-consuming computations and makes it challenging to satisfy engineering design requirements (Morales-Hernández et al., 2023; Hu et al., 2022). Therefore, more effective design decision optimization solutions remain necessary to support the construction industry's transition toward low-carbon, intelligent, and green development strategies.

Multi-Objective Optimization (MOO) issues have attracted significant interest in the domains of computational

resource scheduling and building design. Addressing the dual optimization of building life cycle costs and carbon emissions, Xue et al. (2022) proposed a simulation-based MOO method. The study modeled a passive house in a severely cold climate, established response relationships using artificial neural networks, and identified optimal solutions using the non-dominated sorting genetic algorithm II (NSGA-II). To address workflow scheduling concerns in multi-cloud computing, Mohammadzadeh and Masdari (2023) integrated seagull and grasshopper algorithms, incorporating chaotic mapping to enhance convergence. Results demonstrate superior performance in coverage and other metrics compared to advanced existing techniques. Xue et al. (2023) found that most of the deep neural network methods focus solely on accuracy and ignore efficiency. Therefore, this study proposed a multi-objective (MO) evolutionary algorithm based on a probabilistic stack. Their approach employed an agent model for network construction and introduced a unique cross-mutation methodology, encoding network topology in terms of nearby tables. Experimental results showed 73.6% accuracy on ImageNet. A MO marine predator algorithm based on elite non-dominated sorting and crowded distance mechanism was suggested by Jangir et al. (2023) for MOO issues in order to model the behavior of predator-prey interactions. The algorithm outperformed multiple mainstream algorithms across 32 test problems with diverse features.

Multi-objective particle swarm optimization (MOPSO) is applied across numerous fields of society. Zhang et al. (2025) proposed an MOPSO algorithm incorporating Q-learning to address the high complexity and EC in distributed flow shop scheduling. Their method introduced a particle grouping strategy to accelerate convergence toward the multi-region Pareto frontier. Results demonstrated superior performance over conventional evolutionary algorithms in both solution quality and convergence speed (CS). To address the limited search performance brought on by the fixation of the global optimal selection in MOPSO, Han et al. (2022) proposed an adaptive candidate estimation-assisted MOPSO method. This method enhanced the algorithm's ability to balance convergence and diversity by introducing two evaluation distances. Wang et al. (2023) developed an inflection point-based collaborative MOPSO algorithm for handling mixed variables and multiple constraints in heterogeneous UAV collaborative multitasking. Their algorithm introduced an inflection point learning mechanism to optimize the external profile update. The results revealed that the algorithm could efficiently obtain high-quality allocation schemes that satisfy complex constraints in multiple instances. Agajie et al. (2023) suggested using the MOPSO approach to optimize the capacity and location of distributed power sources. revealed that the MOPSO could effectively reduce the system loss, in which the power loss is reduced by 81.77%. The System performance further improved when the penetration rate of distributed power supply was increased from 10% to 40% under the premise of satisfying the IEEE standard.

In summary, existing research focuses on applying MOO algorithms in the fields of architectural design, scheduling optimization, and neural network construction, with numerous advanced methods proposed. Although MOPSO has achieved significant results in industrial manufacturing, cloud computing scheduling, and UAV task allocation, its practical application in MO decision optimization for building design remains relatively limited. This limitation arises because architectural design problems typically involve various high-dimensional decisions that lead to dimensional disasters. Additionally, each iteration requires high-cost energy simulation software, placing extremely high demands on the CS and computational efficiency of the algorithm. Traditional MOPSO algorithms struggle to meet practical engineering design cycle requirements. This research aims to enhance MOPSO by addressing the premature convergence caused by redundant control parameters and high-dimensional decision space interference in architectural design optimization. To this end, the study constructs a sustainable building design decision model with CL and building EC as dual optimization objectives. Meanwhile, an improved backbone BMOPSO algorithm is introduced to solve the model. The innovation of the study is to introduce the backbone guidance mechanism and adaptive perturbation mechanism into MOPSO to serve the MOO of architectural design. This approach effectively prevents premature convergence in traditional MOPSO caused by control parameter redundancy. Furthermore, integrating comfort and energy consumption objectives enhances the practical applicability of design decisions.

2. Methods and Materials

The study establishes the construction of an MOO system in MO architectural design decision optimization for sustainable buildings. Using CL and EC as key indicators, this study constructs an MOO model containing 12 decision variables constructed by translating complex EC factors into quantifiable parameters via EnergyPlus software. The issue of classic MOPSO's propensity to settle into local optimal states is addressed with the introduction of BMOPSO. Moreover, BMOPSO is optimized by introducing an adaptive perturbation mechanism to form an improved BMOPSO algorithm.

2.1. Multi-Objective Architectural Design Decision Optimization Model

The primary focus in sustainable building design decision optimization is constructing an efficient MOO system. First, a building model for MOO must be developed. The model construction primarily considers two indicators, CL and EC. Increased building EC often correlates with improved CL. The influence mechanism of sustainable building EC is extremely complex, as shown in Fig. 1. On the one hand, the factors affecting building EC are not only related to window parameters, room orientation, envelope properties, indoor environmental parameters, and air-conditioning system settings. These factors exhibit significant coupling effects between the factors. For instance, the thermal insulation performance of the exterior wall and the solar heat gain synergistically affect the cooling and heating EC of the building. On the other hand, there is diversity in the functions of office and residential buildings, the differences in the work schedules, and the focus on the EC needs in cold regions and the hot-summer/cold-winter regions. This leads to significant differences in the patterns of EC influences (Najjar et al., 2022; Ding et al., 2022).

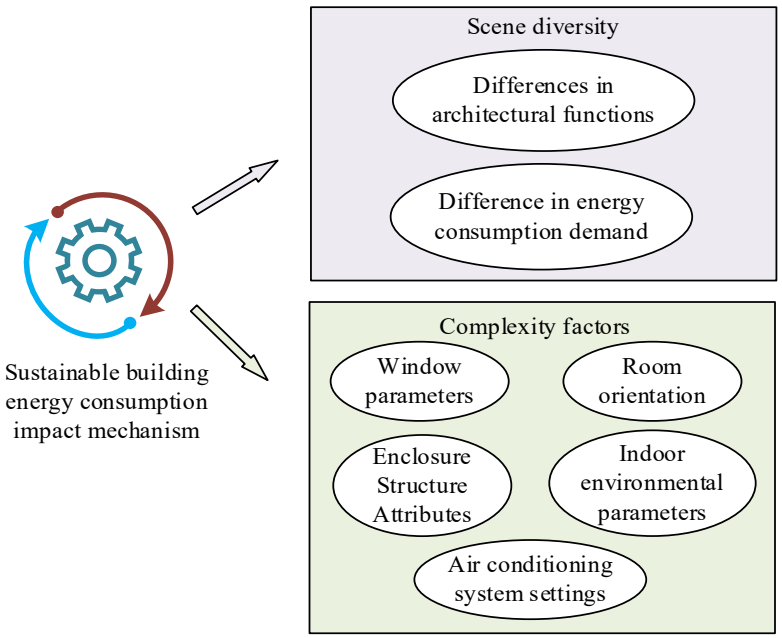

Fig. 1. Impact mechanism of sustainable building energy consumption

Among building modeling simulation software, EnergyPlus is widely used due to its ability to fine-tune the simulation of building thermal processes and energy systems. The software is able to accurately simulate the EC dynamics of buildings under real-time meteorological conditions. Therefore, this study transforms complex building EC influences into quantifiable input parameters. Based on the simulation output of the full EC cycle of a building, a MOO model is constructed to consider both CL and EC. In the MO architectural design decision optimization for sustainable building models, it is necessary to clarify the correlation logic between specific decision variables and the objective function (OF). In the specific operation, the study selects 12 key system parameters as decision variables, as shown in Fig. 2.

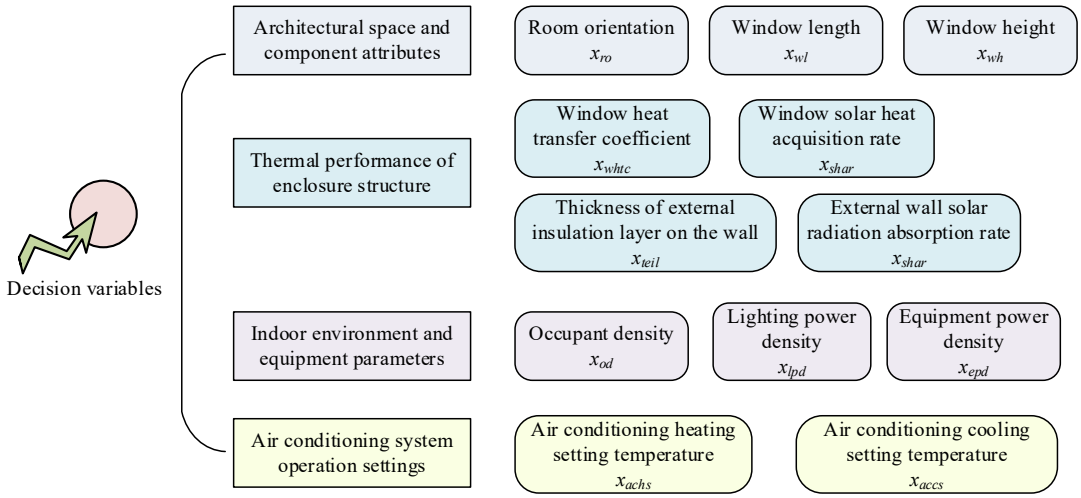

Fig. 2. Decision variables in MOO problems

In Fig. 2, the decision variables are generally categorized into five categories: building space and component properties, envelope thermal performance, indoor environment, equipment parameters, and air-conditioning system operation settings. At the level of building space and component properties, the room orientation x_{ro} mainly affects the solar heat gain and natural ventilation efficiency. Window length x_{wl} and width x_{wh} determine the window-to-wall ratio and daylight area, which affect the building EC and indoor illumination. The window's thermal insulation performance is reflected in the window heat transfer coefficient (x_{whtc}). Lower values indicate higher insulation at the envelope's thermal performance level. The window solar heat gain rate x_{shar} measures the proportion of solar radiation entering the interior through the window. The thickness of the exterior insulation layer, x_{teil}, affects the thermal performance of exterior walls. The external wall solar absorptivity x_{shar} affects solar radiation absorption, where high summer absorptivity increases wall heat gain. At the indoor environmental and equipment parameters level, the occupant density x_{od} determines the indoor heat and moisture load. The lighting power density x_{lpd} and equipment power density x_{epd} reflect the indoor electric equipment heat dissipation and EC, directly affecting the total building EC. At the air conditioning (AC) system level, the AC heating setting temperature x_{achs} and AC cooling setting temperature x_{accs} control the temperature for winter heating and summer cooling. This affects EC and indoor comfort (Dalirazar et al., 2022; Usman et al., 2023). In addition, the material type can be indirectly quantified through the thermal performance parameters of the enclosure structure. Therefore, it does not need to be listed separately as a variable. Occupant schedules did not include dynamic variables as the experimental scenario has

been standardized. After assigning specific values to each parameter, the EnergyPlus building EC simulation software executes the simulation calculations. Based on the input parameters, the software can simulate the real-time energy flow and indoor thermal environment dynamics of the building throughout the year. The simulation outputs, the total annual EC of the building and the annual comfort hours in a year are output as the quantitative results of CL and EC, respectively. The calculation expression of CL is shown in Eq. (1).

$$CL = \sum_{a=1}^{A} \Delta t_a \tag{1}$$

Calculation of comfort follows the internationally recognized ASHRAE 55 standard. In Eq. (1), A is the total number of all time steps in a year. Δt_a represents the duration of the a th time step. Its value depends on whether the indoor environmental parameters at the a th time step falls within the comfort zone. If temperature, humidity, and wind speed (WS) at the a th time step remains within the comfort interval, then $\Delta t_a = 1$. Otherwise, $\Delta t_a = 0$. The comfort interval follows ASHRAE 55 standard, which states that the comfort range for indoor temperatures is 20°C to 26°C. The comfort range for Relative Humidity (RH) is 30% to 70%, and the comfort range for Wind Speed (WS) is 0.1 m/s to 0.2 m/s. Eq. (2) shows the EC calculation.

$$EC = \sum \left[E_{heat}(t) + E_{cool}(t) + E_{light}(t) + E_{vent}(t) + E_{equip}(t) \right] \tag{2}$$

In Eq. (2), $E_{heat}(t)$, $E_{cool}(t)$, $E_{light}(t)$, $E_{vent}(t)$, and $E_{equip}(t)$ denote heating, cooling, lighting, ventilation, and equipment system EC, respectively. The resulting multi-objective optimization model is shown in Eq. (3).

$$\begin{cases} \min F = (-CL(X), EC(X)) \\ s.t. X = (x_{ro}, x_{wl}, x_{wh}, x_{whtc}, x_{shar}, x_{teil}, x_{shar}, x_{od}, x_{lpd}, x_{epd}, x_{achs}, x_{accs}) \end{cases} \tag{3}$$

In Eq. (3), X denotes the decision variable.

2.2. Multi-Objective Architectural Design Decision Optimization for Sustainable Buildings Based on Improved PSO Algorithms

2.2.1. Principles and Iterative Mechanism of Basic PSO Algorithms

After constructing the multi-objective architectural design decision model for sustainable buildings, the study proceeds to solve it. Given its fast convergence speed and strong global search capability, the study employs the PSO algorithm for model solution. PSO efficiently explores complex decision space in the MOO model by simulating collaborative foraging behavior in bird flocks. Its iteration mechanism effectively balances global exploration and local optimization (Akay et al., 2022). The particle velocity update formula in PSO appears in Eq. (4).

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot r_1 \cdot (pBest_i - x_i^t) + c_2 \cdot r_2 \cdot (gBest - x_i^t) \tag{4}$$

In Eq (4), v_i^t and v_i^{t+1} represent the velocity vectors of the particle i at the t th and $t+1$ th iterations, respectively. ω is the IW, which balances the effect of the particle's previous velocity on the current iteration. c_1 and c_2 are learning factors that control the step size of the particle learning toward the individual optimal position (OP) $pBest_i$ and the global OP $gBest$, respectively. r_1 and r_2 are random numbers in the interval [0,1]. Eq. (5) displays the position update expression.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{5}$$

In Eq. (5), x_i^t and x_i^{t+1} denote the position vectors of the particle i at the t th and $t+1$ th iterations, respectively. The schematic diagram of particle iteration in the PSO algorithm is shown in Fig. 3.

In the planar rectangular coordinate system of Fig. 3, the current position of the particle is x^t, and the current velocity is v^t. When iterating, the IW term ωvt maintains the original motion trend of the particle. The individual cognitive term $c_1 \cdot r_1 \cdot (pBest - x^t)$ drives the particle to move toward its own historical OP. The social $pBest$ -cognitive term $c_2 \cdot r_2 \cdot (gBest - x^t)$ guides the particle toward the global OP $gBest$ of the population. The superposition of the three vectors obtains the new velocity v^{t+1} and determines the new position x^{t+1}. The optimal solution is gradually approximated in this way.

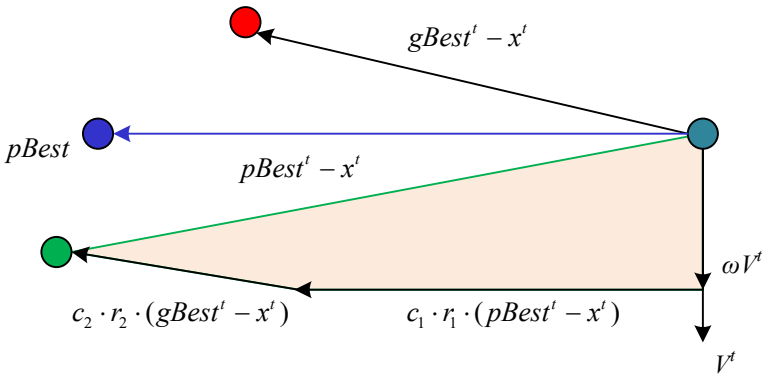

Fig. 3. Schematic diagram of particle iteration in PSO algorithm

2.2.2. BMOPSO Improvement and Adaptive Disturbance Mechanism Design

However, there are more control parameters in the traditional PSO algorithm, which easily falls into local optima (Lu et al., 2023; Guo et al., 2022). Therefore, this study introduces the BMOPSO algorithm, which reduces the control parameters of the original algorithm. Understanding the single-objective particle update formula is a prerequisite to comprehending the MO formula. The backbone particle swarm optimization (BPSO) particle position update (PPU) calculation for a single objective problem is shown in Eq. (6).

$$x_{i,j}(t+1) = N(\frac{pBest_{i,j}(t) + gBest_j(t)}{2}, \left|pBest_{i,j}(t) - gBest_j(t)\right|) \tag{6}$$

In Eq. (6), $x_{i,j}(t+1)$ denotes the new position of the particle i on the jth decision variable in the $t+1$th iteration. $N(\mu,\sigma)$ denotes the Gaussian distribution function. Among them, μ is the mean and σ is the variance. Since the BPSO algorithm can be derived from PSO, a variant of Eq. (6) is thus obtained as shown in Eq. (7).

$$x_{i,j}(t+1) = \begin{cases} N(\frac{pBest_{i,j}(t) + gBest_j(t)}{2}, |\, pBest_{i,j}(t) - gBest_j(t)\,|), & U(0,1) < 0.5 \\ pBest_{i,j}(t), & otherwise \end{cases} \tag{7}$$

In Eq. (7), $U(0,1)$ is a uniformly distributed random number on the interval [0, 1]. Then it is extended to MO problem solving to obtain the BMOPSO algorithm. The PPU expression of this algorithm is displayed in Eq. (8).

$$x_{i,j}(t+1) = \begin{cases} N\left(\frac{r_3 \times pBest_{i,j}(t) + (1-r_3) \times gBest_j(t)}{2}, |\, pBest_{i,j}(t) - gBest_{i,j}(t)\,|\right), & U(0,1) < 0.5 \\ gBest_{i,j}(t), & otherwise \end{cases} \tag{8}$$

In Eq. (8), r_3 denotes a random number in [0, 1]. When $pBest_{i,j}(t)$ approaches $gBest_{i,j}(t)$. The normal distribution sampling result approaches the global optimum. However, this causes particles to persistently search near known regions, limiting exploration. To address this, an enhanced BMOPSO method incorporating an adaptive perturbation mechanism is introduced. The adaptive perturbation mechanism triggers automatically when search stagnation is detected. This perturbation pushes particles from current regions to explore new areas and does so to explore new possibilities. Disturbance magnitude adapts dynamically. When particle diversity is low, amplitude increases to enhance the exploration; when high, it decreases. The PPU for this algorithm appears in Eq. (9).

$$x_{i,j}(t+1) = \begin{cases} N\left(\frac{r_3 \times pBest_{i,j}(t) + (1-r_3) \times gBest_j(t)}{2}, |\, pBest_{i,j}(t) - gBest_{i,j}(t)\,| + \beta_j\right), & U(0,1) < 0.5 \\ gBest_{i,j}(t), & otherwise \end{cases} \tag{9}$$

In Eq. (9), β_j denotes the perturbation factor. Its value depends on the similarity between the global extreme points and the individual extreme points, as shown in Eq. (10).

$$\beta_j = \begin{cases} (x_j^{\max} - x_j^{\min}) \times e^{(-5t/T)}, & pro_d \geq rand \\ 0, & otherwise \end{cases} \tag{10}$$

In Eq. (10), $x_j^{\max}$ and $x_j^{\min}$ display the maximum and minimum values taken by the j th decision variable. T is the maximum iteration. pro_d is the disturbance probability, as shown in Eq. (11).

$$pro_d = 0.5 \times (1 - \frac{1}{Q} \sum_{q=1}^{Q} \left| \frac{f_m(PBest_i(t)) - f_m(GBest_i(t))}{f_m^{\max} - f_m^{\min}} \right| \tag{11}$$

In Eq. (11), $PBest_i(t)$ is the individual extreme value point and $f_q(PBest_i(t))$ is the q th OF value of the point. $GBest_i(t)$ is the global extreme value point, $f_q(GBest_i(t)$ is the q th OF value of the point. $f_q^{\max}$ and $f_q^{\min}$ denote the maximum and minimum values about the q th OF value, respectively. Q denotes all the OF values. Fig. 4 depicts the flow of the enhanced BMOPSO algorithm.

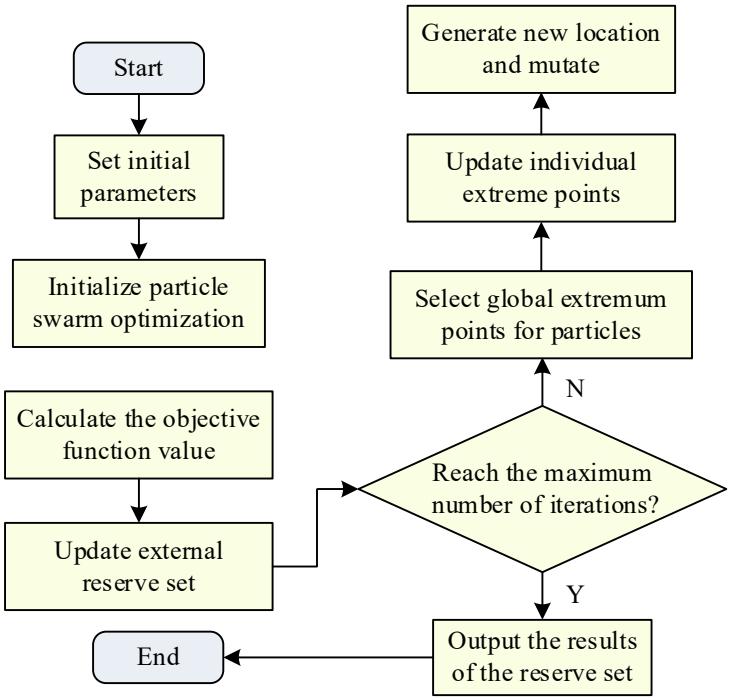

Fig. 4. Flowchart of improved BMOPSO algorithm

Initializing the parameters and particle population is the first step in the enhanced BMOPSO method shown in Fig. 4. It then iteratively optimizes building EC and CL by calculating the value of the objective functions, updating the external archive, selecting the global and individual best positions, and updating the particle positions. The algorithm terminates upon reaching the maximum iteration, outputting the non-dominated solutions from the external archive as the final result.

2.3. Experimental Platform Setup and Implementation Process

To realize MO architectural design decision optimization for sustainable buildings, the research integrates EnergyPlus, Matlab, and Visual C++ to complete the platform construction of the improved BMOPSO algorithm. The specific implementation process appears in Fig. 5.

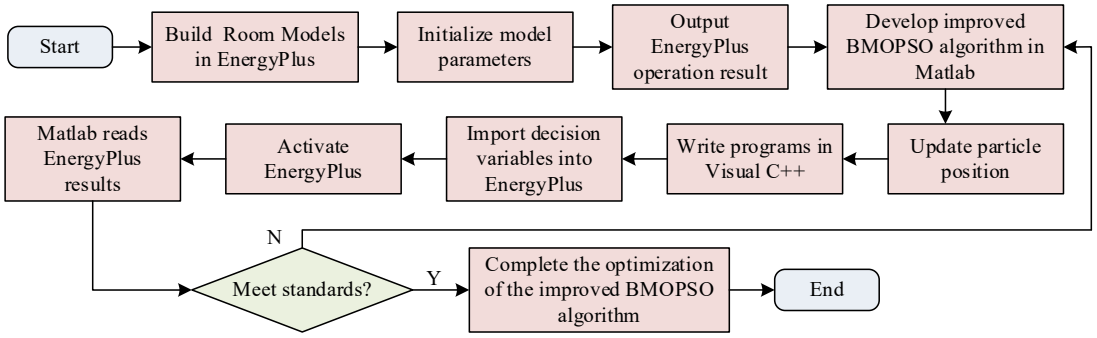

Fig. 5. Platform construction for improving BMOPSO algorithm

In Fig. 5, the room model is first constructed in EnergyPlus, where model parameters are initialized, and simulation results are generated. Next, the improved BMOPSO algorithm is implemented in Matlab, while a Visual C++ program handles particle position updates. Updated particle positions are imported into EnergyPlus, which activates the software

and inputs the decision variables. After EnergyPlus is run, Matlab reads the results and determines if the criteria are met. If satisfied, the optimization is completed. If not satisfied, the process returns to updating particle positions, and iterations continue until an optimal solution is found.

3. Results

To validate the improved BMOPSO algorithm's effectiveness for MO architectural design decision optimization, two building scenarios are analyzed: a single office and a three-room residence. These scenarios differ significantly in terms of functional requirements, personnel density, and equipment usage patterns. They represent major commercial and residential building types, effectively testing algorithm adaptability under varying constraints. Performance metrics include the spatial coverage (SC) measure, hypervolume (HV) strategy, running time (RT), and Pareto frontiers, which are used as performance evaluation metrics for each algorithm. Finally, CL and EC optimization results are compared for the BMOPSO algorithm before and after improvements.

3.1. Experimental Scenario and Environment Design

The study uses the suggested algorithm in two construction situations to confirm its efficacy. Scenario 1 is a single-room office building, and Scenario 2 is a three-room residential building. The styles of different scenario buildings are shown in Fig. 6.

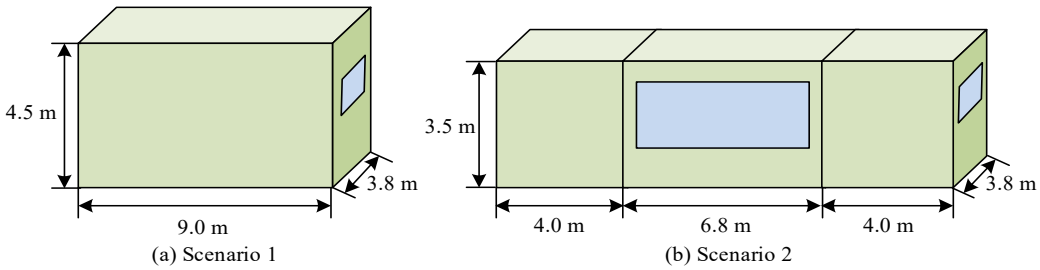

Fig. 6. Styles of buildings in different scenarios

Table 1 displays the range of values for each choice variable.

Table 1. Range and initial values of 12 decision variables in multi-objective optimization model

Decision variables	Range	Initial value
xro (°)	[0, 360]	180
xwl (m)	(0.8, 2.5)	1.5
xwh (m)	(0.8, 2.8)	1.4
xwhtc [w/(m2·k)]	(1.8, 4.5)	2.8
xshar	(0.2, 0.6)	0.4
xteil (m)	(0.03, 0.08)	0.05
xshar	(0.3, 0.8)	0.5
xod (persons/m2)	(0.15, 0.3)	0.25
xlpd (w/m2)	[5, 10]	7
xepd (w/m2)	[8, 15]	12
xachs (℃)	[18, 22]	20
xaccs (℃)	[24, 27]	25

Table 2 displays the experimental hardware environment.

Table 2. Experimental hardware environment configuration parameters of optimized BMOPSO algorithm

Name	Setup
Operating system	Windows 10 64 bit
CPU	I7-8750H, Six core, 2.20GHz
Memory	8GB
Hard disk	1T
GPU	NVIDIA GeForce GTX 1050Ti

3.2. Comparative experimental validation

The study compares several popular MOO algorithms, such as MOPSO, NSGA-II, and MO artificial hummingbird

algorithm (MOAHA), to confirm the superiority of the suggested enhanced BMOPSO algorithm in MO architectural design optimization for sustainable buildings. The maximum iteration count is set to 100, and the population size is 100. Spacing SC, HV strategy, RT, and Pareto frontier are selected as performance evaluation metrics of each algorithm. Each algorithm executes ten times, and the average value is used as the final result to ensure the reliability of the results. Preliminary testing confirms 100 iterations for all algorithms to reach a convergence state. Fig. 7 displays each algorithm's HV values across various conditions. The average HV value of the enhanced BMOPSO method in Scenario 1 of Fig. 7(a) is 29963, but the NSGA-II is only 19246, a number that is much lower than that of the enhanced BMOPSO algorithm. In Fig. 7(b), the average HV of the improved BMOPSO algorithm is as high as 42639 in Scenario 2, while MOPSO and MOAHA are only 14628 and 15639, respectively. The improved algorithm maintains the highest HV across scenarios with minimal performance fluctuation between scenarios. This demonstrates strong robustness and is suitable for diverse building scenarios. Other algorithms show substantial inter-scenario fluctuations, exposing high sensitivity to scenarios.

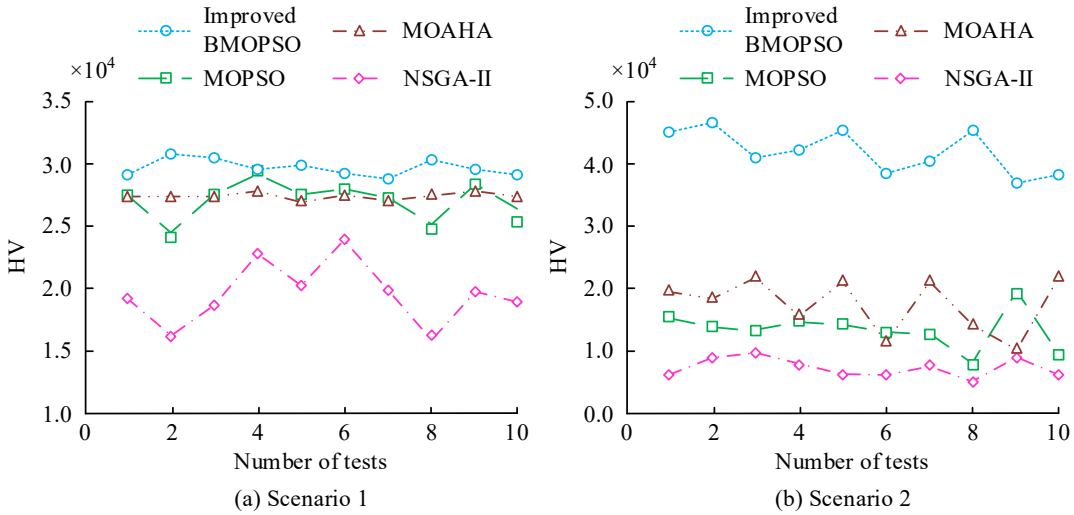

Fig. 7. HV values of various algorithms in different scenarios

The study further compares SC values of each algorithm in different scenarios. The results are shown in Table 3. SC (#Improved BMOPSO) indicates the percentage where comparison algorithms dominate Improved BMOPSO. SC (Improved BMOPSO, #) indicates where improved BMOPSO dominates comparison algorithms. In Table 3, the SC mean value of the improved BMOPSO algorithm with respect to NSGA-II is as high as 1.000 in Scenarios 1 and 2. Meanwhile, the SC mean value of NSGA-II with respect to the improved algorithm is 0. This indicates that all the solutions of the improved algorithm completely dominate the solutions of NSGA-II. SC mean values for the improved algorithm are 0.364 and 0.361 for MOPSO and 0.314 and 0.386 for MOAHA in different scenarios. Whereas the SC mean values of MOPSO against the improved algorithm are 0.089 and 0.281. The SC mean values of MOAHA against the improved algorithm are 0.173 and 0.151, which are lower than the corresponding values for the improved algorithm against MOPSO and MOAHA. This demonstrates the enhanced algorithm's overall convergence superiority over the three comparable methods.

The study further compares the runtime of each algorithm in different scenarios. In Fig. 8(a), Scenario 1, the average runtime of the improved BMOPSO algorithm is only 1.38h, whereas the MOAHA algorithm has the longest runtime, with a high average of 1.83h. The NSGA-II has an average runtime of 1.64h, which is higher than that of the improved BMOPSO algorithm by 0.26h and lower than that of the MOPSO algorithm by 1.73h. In Fig. 8(b), in Scenario 2, the average runtime of the improved BMOPSO algorithm is only 3.38h, whereas the NSGA-II reaches an average of 5.58h. It exceeds the 4.51-hour time of the MOPSO algorithm. At 6.03 hours, the MOAHA algorithm continues to have the longest runtime. This shows that the improved BMOPSO algorithm proposed by the study maintains the minimum runtime in different scenarios compared to the rest of the algorithms. The method effectively enhances both time efficiency and scene robustness.

Table 3. SC measurement values of various algorithms in different building scenarios

Scenario	Algorithm	SC (Improved BMOPSO, #)				SC (#, Improved BMOPSO)			
		Best	Worst	Average	Std	Best	Worst	Average	Std
Scenario 1	NSGA-II	1.000	1.000	1.000	0.000	0.000	1.000	1.000	1.000
	MOPSO	0.437	0.291	0.364	0.060	0.154	0.024	0.089	0.056
	MOAHA	0.349	0.279	0.314	0.024	0.217	0.129	0.173	0.031
Scenario 2	NSGA-II	1.000	1.000	1.000	0.000	0.000	1.000	1.000	1.000
	MOPSO	0.421	0.301	0.361	0.045	0.301	0.261	0.281	0.016
	MOAHA	0.401	0.315	0.386	0.066	0.191	0.111	0.151	0.033

The study further compares algorithm Pareto frontiers across scenarios, as shown in Fig. 9. In both scenarios, the improved BMOPSO Pareto frontier lies closer to the low-EC/high-CL region with a more uniform distribution. It shows clear advantages in balancing the building EC and comfort. NSGA-II shows a relatively inferior and dispersed frontier struggling to balance EC and comfort. The MOPSO and MOAHA algorithms perform better than the NSGA-II, but are still inferior to the BMOPSO algorithm. This demonstrates how the backbone guidance and adaptive perturbation mechanism enhance convergence accuracy and solution diversity. The approach efficiently identifies optimal EC-comfort trade-offs in complex building scenarios.

The study concludes with EC and CL optimization tests for buildings across scenarios, comparing the unimproved BMOPSO algorithm with the improved BMOPSO version. As iterations progress, the EC value decreases while the CL value increases. In Fig. 10(a), in Scenario 1, the BMOPSO algorithm converges around 40 iterations, and the CL value stabilizes at 4351h. The improved BMOPSO algorithm, on the other hand, converges only around 20 iterations, and the CL value stabilizes at 4836h. In Fig. 10(b), the EC value of the improved BMOPSO algorithm is finally stabilized at 625kWh, which is a decrease of 3475kWh compared to the BMOPSO algorithm. In Figs. 10(c) and 10(d), in Scenario 2, the CL value of the improved BMOPSO algorithm is finally stabilized at 7625h, which is an increase of 693h compared to the BMOPSO algorithm. Meanwhile, the improved algorithm's EC value is finally stabilized at 1953kWh, while the BMOPSO algorithm is as high as 2369kWh. It indicates that the improved BMOPSO algorithm achieves a better balance between the dual objectives of user comfort and building energy efficiency.

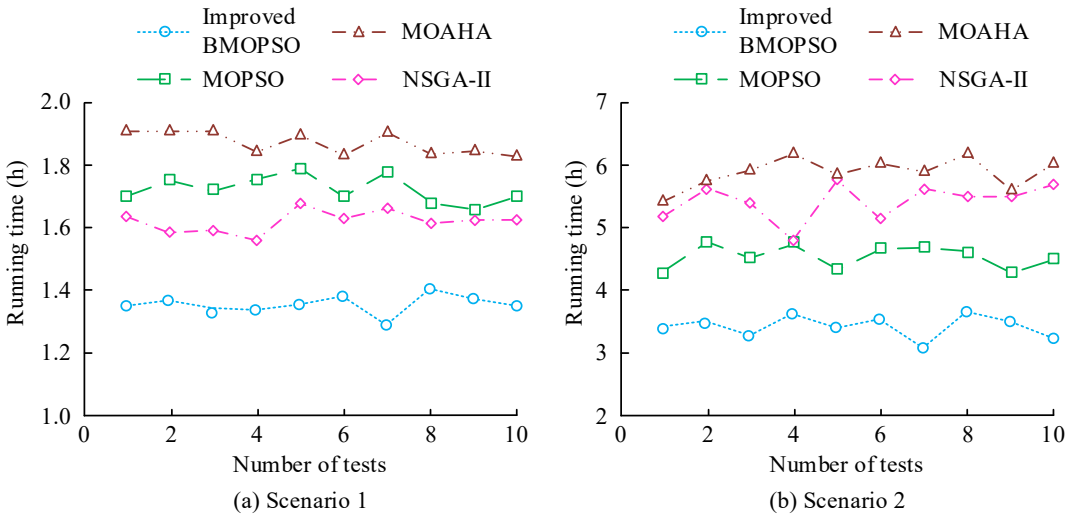

Fig. 8. Runtime of various algorithms in different scenarios

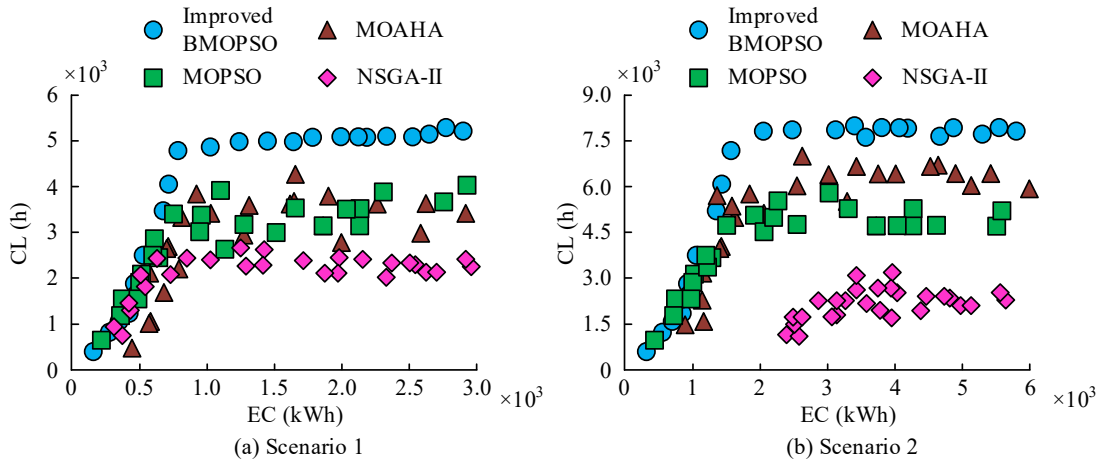

Fig. 9. Pareto frontiers in different scenarios

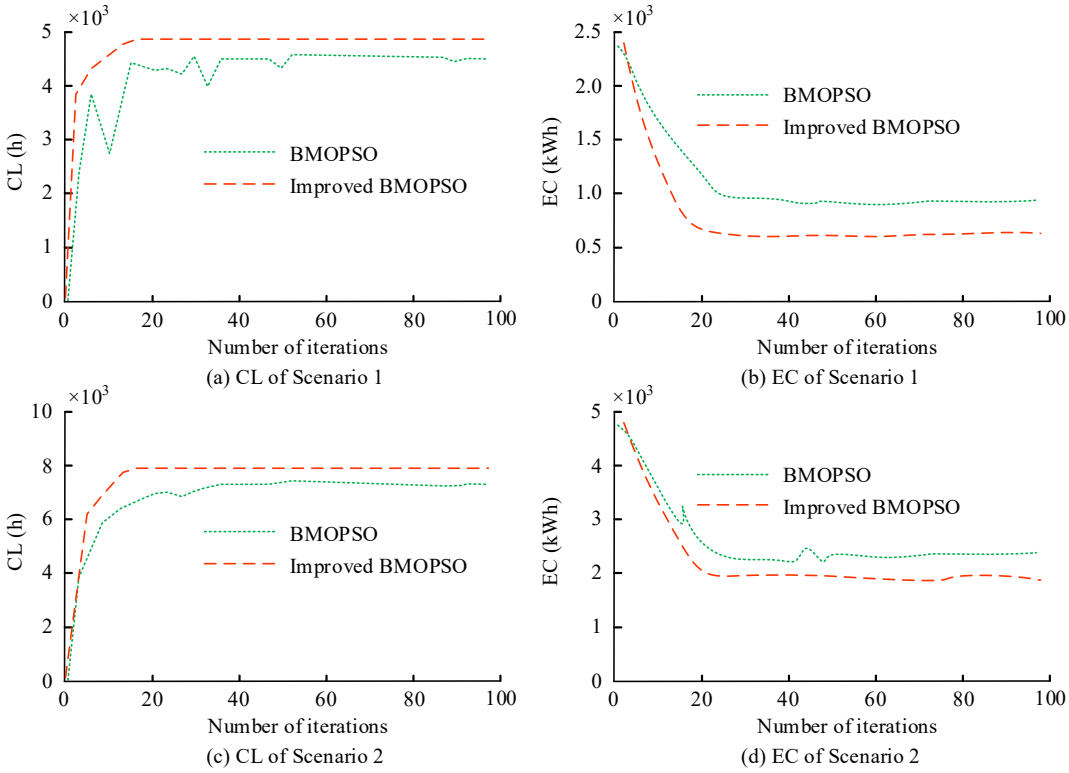

Fig. 10. EC and CL optimization testing

4. Discussion and Conclusion

The study proposed an improved BMOPSO algorithm for MO architectural design decision optimization for sustainable buildings. Furthermore, its effectiveness was verified through various experiments. Regarding HV values, the enhanced BMOPSO method in Scenario 1 had an average HV value of 29963, which was noticeably higher than NSGA-II's (19246). In Scenario 2, the average HV value was as high as 42639, which was much higher than that of MOPSO (14628) and MOAHA (15639), with minimal fluctuation between scenarios. As the improved algorithm optimizes the search strategy, it enhanced the robustness of MOO and adapted to different building scenarios, while the comparison algorithms were highly sensitive to the scenarios. The SC value results revealed that the SC mean value of the improved BMOPSO algorithm with respect to NSGA-II was 1.000, which completely dominated the NSGA-II solution. The SC mean values for MOPSO and MOAHA were also significantly higher than the comparison algorithms. This stemmed from better convergence of the solution set and was able to approximate the optimal solution more efficiently in the search. In terms of RT, the average runtime of improved BMOPSO was 1.38h in Scenario 1 and 3.38h in Scenario 2, both of which remain the lowest. This resulted from dynamic particle updates and other strategies to reduce invalid computation and improve time efficiency. The comparison of Pareto frontiers indicated that the frontiers of the improved BMOPSO algorithm were closer to the low EC and high comfort areas and uniformly distributed, which enabled a better balance between the EC and comfort of the building. The primary cause was that the improved algorithm enhanced the global search and MO co-optimization capabilities, whereas the conventional method was prone to slipping into the local optima. In the EC and CL optimization test, the improved algorithm converged faster, achieving a higher CL value and a lower EC value. In Scenario 1, the improved BMOPSO algorithm converged in about 20 iterations, and the CL value was stabilized at 4,836 hours. In Scenario 2, the EC stabilized at 1,953 kWh, achieving a better balance between user comfort and building energy savings. In summary, the improved BMOPSO algorithm demonstrates better performance in MO architectural design decision optimization for sustainable buildings. The research results have significant practical value. This algorithm can be encapsulated as a plugin and integrated into existing building information models or computer-aided design software. In the early stages of designing a scheme, designers only need to input preliminary design parameter ranges to quickly obtain a Pareto frontier solution set that optimizes the balance between EC and comfort. Designers can intuitively see the impact of adjusting a parameter on two objectives through a visual interface, thereby making more efficient decisions.

However, the study only tests two scenarios: a single office and three residential buildings, and does not involve complex scenarios such as high-rise buildings and commercial complexes. In the future, it is necessary to expand the types of scenarios to enhance the versatility of the algorithm. Additionally, the simulation results of building EC and comfort rely on the built-in model of EnergyPlus. Deviations between actual parameters and software assumptions in actual engineering may affect the accuracy of the optimization results. Therefore, the simulation model should be calibrated based on monitoring data from actual buildings.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Reference

Agajie, T. F., Gebru, F. M., Salau, A. O., and Aeggegn, D. B. (2023). Investigation of distributed generation penetration limits in distribution networks using multi-objective particle swarm optimization technique. Journal of Electrical Engineering & Technology, 18(6), 4025–4038. https://doi.org/10.1007/s42835-023-01457-4

Akay, B., Karaboga, D., and Akay, R. (2022). A comprehensive survey on optimizing deep learning models by metaheuristics. Artificial Intelligence Review, 55(2), 829–894. https://doi.org/10.1007/s10462-021-09992-0

Dalirazar, S. and Sabzi, Z. (2022). Barriers to sustainable development: Critical social factors influencing the sustainable building development based on Swedish experts' perspectives. Sustainable Development, 30(6), 1963–1974. https://doi.org/10.1002/sd.2362

Ding, Y., Pang, Z., Lan, K., Yao, Y., Panzarasa, G., Xu, L., and Hu, L. (2022). Emerging engineered wood for building applications. Chemical Reviews, 123(5), 1843–1888. https://doi.org/10.1021/acs.chemrev.2c00450

Guo, C., Chen, X., Li, Q., Ding, G., Yue, H., and Zhang, J. (2022). Milling optimization of GH4169 nickel-based superalloy under minimal quantity lubrication condition based on multi-objective particle swarm optimization algorithm. The International Journal of Advanced Manufacturing Technology, 123(11), 3983–3994. https://doi.org/10.1007/s00170-022-10461-3

Han, H. G., Zhang, L. L., Hou, Y., and Qiao, J. (2022). Adaptive candidate estimation-assisted multi-objective particle swarm optimization. Science China Technological Sciences, 65(8), 1685–1699. https://doi.org/10.1007/s11431-021-2018-x

Hu, Z., Shaloudegi, K., Zhang, G., and Yu, Y. (2022). Federated learning meets multi-objective optimization. IEEE Transactions on Network Science and Engineering, 9(4), 2039–2051. https://doi.org/10.1109/TNSE.2022.3169117

Jangir, P., Buch, H., Mirjalili, S., and Manoharan, P. (2023). MOMPA: Multi-objective marine predator algorithm for solving multi-objective optimization problems. Evolutionary Intelligence, 16(1), 169–195. https://doi.org/10.1007/s12065-021-00649-z

Lu, Z., Cheng, R., Jin, Y., Tan, K. C., and Deb, K. (2023). Neural architecture search as multi-objective optimization benchmarks: Problem formulation and performance assessment. IEEE Transactions on Evolutionary Computation, 28(2), 323–337. https://doi.org/10.1109/TEVC.2022.3233364

Mangalampalli, S., Swain, S. K., and Mangalampalli, V. K. (2022). Multi-objective task scheduling in cloud computing using cat swarm optimization algorithm. Arabian Journal for Science and Engineering, 47(2), 1821–1830. https://doi.org/10.1007/s13369-021-06076-7

Mohammadzadeh, A. and Masdari, M. (2023). Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm. Journal of Ambient Intelligence and Humanized Computing, 14(4), 3509–3529. https://doi.org/10.1007/s12652-021-03482-5

Morales-Hernández, A., Van Nieuwenhuyse, I., and Rojas Gonzalez, S. (2023). A survey on multi-objective hyperparameter optimization algorithms for machine learning. Artificial Intelligence Review, 56(8), 8043–8093. https://doi.org/10.1007/s10462-022-10359-2

Najjar, M. K., Figueiredo, K., Evangelista, A. C. J., Hammad, A. W., Tam, V. W., and Haddad, A. (2022). Life cycle assessment methodology integrated with BIM as a decision-making tool at early-stages of building design. International Journal of Construction Management, 22(4), 541–555. https://doi.org/10.1080/15623599.2019.1637098

Ramzanpoor, Y., Hosseini Shirvani, M., and Golsorkhtabaramiri, M. (2022). Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. Complex & Intelligent Systems, 8(1), 361–392. https://doi.org/10.1007/s40747-021-00368-z

Usman, A. M., and Abdullah, M. K. (2023). An assessment of building energy consumption characteristics using analytical energy and carbon footprint assessment model. *Green and Low-Carbon Economy, 1*(1), 28–40. https://doi.org/10.47852/bonviewGLCE3202545

Wang, F., Huang, Z. L., Han, M. C., Xing, L., and Wang, L. (2023). A knee point based coevolution multi-objective particle swarm optimization algorithm for heterogeneous UAV cooperative multi-task allocation. Acta Automatica Sinica, 49(2), 399–414. https://doi.org/10.16383/j.aas.c210696

Xue, Q., Wang, Z., and Chen, Q. (2022). Multi-objective optimization of building design for life cycle cost and CO_2 emissions: A case study of a low-energy residential building in a severe cold climate. Building Simulation, 15(1), 83–98. https://doi.org/10.1007/s12273-021-0796-5

Xue, Y., Chen, C., and Słowik, A. (2023). Neural architecture search based on a multi-objective evolutionary algorithm with probability stack. IEEE Transactions on Evolutionary Computation, 27(4), 778–786. https://doi.org/10.1109/TEVC.2023.3252612

Zhang, W., Geng, H., Li, C., Gen, M., Zhang, G., and Deng, M. (2025). Q-learning-based multi-objective particle swarm optimization with local search within factories for energy-efficient distributed flow-shop scheduling problem. Journal of Intelligent Manufacturing, 36(1), 185–208. https://doi.org/10.1007/s10845-023-02227-9

Zhong, W., Siming, M. A., Tingting, W., Tingting, W. A. N. G., Wei, T. A. O., and Jing, X. U. (2024). Research on carbon emission accounting and the "Dual Carbon" transformation path of industrial parks. Southern Energy Construction, 11(5), 191–199. https://doi.org/10.16516/j.ceec.2024.5.20

Song Du earned his Master's degree in History from Nanjing Normal University in China in 2008. He currently serves as the Deputy Director of the Infrastructure Department at Jiangsu Maritime Institute. With 15 years of experience in higher education, his work focuses on educational management, student management, and student ideological education. His primary research interests are now educational management and campus construction management.