Green Building Design Optimization Based on Multi-Objective Particle Swarm Algorithm

Yousheng Yuan

Abstract—**With the development of society and the growth of human demands, energy consumption is also continuously increasing. Green building energy-saving design is currently an important component in the field of energy consumption. It is also one of the current challenges that need to be addressed. To optimize the design of green buildings and reduce energy consumption while ensuring building comfort, a multi-objective green building energy-saving optimization model is constructed. An improved multi-objective back-bone particle swarm optimization algorithm based on adaptive disturbance factors is designed. To reduce the running time, a decomposition-based proxy model assisted multi-objective particle swarm optimization algorithm is designed. A new sample selection strategy guided by dual reserve sets is also designed. According to the experimental results, in single room and three bedroom buildings, the average hypervolume measurement values were 29311 and 49504, respectively. The average hypervolume measurement values of the proxy model were 21153 and 40230, respectively. The designed algorithm has good performance, which can provide technical support for the optimization design of green buildings.**

Index Terms—**Green, Architecture, MOPSO, Design, Optimization**

I. Introduction

ENERGY consumption is constantly increasing. The construction industry accounts for 40%. The energy-saving design has become the main direction of architectural design. Green building energy-saving design is a Multi-Objective Optimization (MOO) problem with multiple conflicting performance indicators [1]. With the advance in computer science, more researchers are starting from models to optimize the energy consumption. Lin et al. aimed to optimize building performance. An interactive architectural performance optimization model was designed. A multi-objective building optimization method was designed [2]. However, the algorithms used in these research also have problems. For example, control parameters are sensitive and the operating cost is expensive [3]. The proxy model assisted evolutionary algorithm can save the computational cost of evolutionary algorithms and solve the expensive operation costs. In previous studies, some scholars have designed an efficient proxy assisted hybrid optimization algorithm to solve expensive problems. This algorithm combines teaching based optimization algorithm and differential evolution algorithm [4]. However, when dealing with MOO problems, this method also faces difficulties in sample selection and complex model construction [5]. The current problems in optimizing green building design include algorithm sensitivity to control parameters and time-consuming calculations.

Manuscript received May 15, 2024; revised May 20, 2025.

Y. S. Yuan is a Lecturer of the School of Civil Engineering and Architecture, Henan University of Science and Technology, Luoyang, 471000, China (e-mail: yysheng19810308@163.com)

To address the sensitive control parameters, a Back-Bone Multi-Objective Particle Swarm Optimization (PSO) based on Adaptive Disturbance Factor (BBMOPSO-A) is innovatively designed firstly. Secondly, in response to the insufficient computation time of evolutionary optimization-based methods, a decomposition-based proxy model assisted multi-objective PSO (MOPSO) is designed, and the generation of proxy model management strategies is also designed to better reduce the running time of the algorithm. There are two innovations. The first point is the combination of disturbance factors and backbone MOPSO. The second point is the combination of proxy models and MOPSO. The research aims to enhance the global optimization ability of MOPSO, and reduce the running time and computational time, providing technical support for green building energy-saving design.

II. Related Works

With the development of society, reducing building energy consumption and improving building energy efficiency have become issues that experts from all over the world consider together. Many experts have conducted research on the design optimization of green buildings. Zhou et al. selected a green office building in a certain city in China as an example to evaluate the indoor environmental quality and energy consumption of green office buildings, providing reference for the design of green buildings. The on-site measurement method was used to obtain the indoor environmental quality. The questionnaire survey was used to obtain users' satisfaction with the building. The green office building was far less than the constraint value of national standards [6]. Almeida et al. selected university green buildings and non-green buildings with similar characteristics to analyze the impact of residential behavior on energy use. The building simulation method is used to compare the energy utilization of buildings. The interaction between occupants and systems in the building was simulated. The research results showed that the impact of occupants on building energy performance was about 72%, which provided reference for the green buildings [7]. Yue et al. designed a three-dimensional urban landscape model to analyze the application of green building materials. The auto-correlation function was used to simulate landscape signals. In addition, the study also adopted fuzzy evaluation methods and improved the 3D model. Different types of green building materials play different functions in urban landscape design [8]. Andiyan et al. analyzed the response of buildings to environmental issues. A harmonious relationship between the main functions of office buildings and the environment was created through the application of green building concepts. The results indicated that the green building concept could continue to be used to address

environmental issues in buildings [9].

MOPSO can effectively solve MOO problems, which has been applied to solve MOO problems in different fields. Trng et al. designed a combination algorithm that combined MOPSO and third-generation non-dominated sorting genetic algorithm to achieve cost optimization of stiffness parameters for powertrain suspension systems. This study transformed the cost optimization into a MOO problem with six optimization objectives. It was better than that of a single MOPSO or the third generation non-dominated sorting genetic algorithm [10]. Xu et al. used probability theory to analyze the global convergence of MOPSO. The study also defined convergence metrics. The global convergence was transformed into the convergence of the metric sequence. The MOPSO did not guarantee global convergence with probability [11]. Anh et al. designed an improved MOPSO for optimal energy management of micro-grids. Pareto frontiers were used to seek multiple objective solutions. The algorithm designed in the research could be optimized in real-time. It had better performance than other heuristic algorithms [12]. Zhi et al. designed a control optimization method based on MOPSO to optimize the condenser control system in nuclear power plants. In this control method, the optimization object was the control parameters. The optimization objective was the step response performance. The designed method could obtain high-quality control parameters, with good performance [13].

In summary, the research on green building optimization and MOPSO is relatively rich. Moreover, the methods and fields involved are also relatively diverse. However, these research also have shortcomings. For example, the convergence of MOPSO is highly dependent on control parameters. Running is time-consuming and has many control parameters. Therefore, an improved multi-objective backbone PSO based on adaptive disturbance factors is designed. A decomposition-based proxy model assisted MOPSO is also developed.

III. Construction of Energy-Saving Design Method for Green Buildings Based on MOPSO

In this chapter, a BBMOPSO-A is constructed to address the shortcomings of traditional PSO. To improve the operational cost of the multi-objective backbone PSO, a decomposition-based proxy model assisted MOPSO is also developed.

A. The BBMOPSO-A for green building energy-saving design

There are two problems with the traditional PSO for the energy-saving design. One is sensitive to the control parameter value. The other is time-consuming in calculation [14-16]. A BBMOPSO-A model is designed to address the sensitivity of control parameter value. For this design, the main indicators are energy consumption and comfort. Therefore, the study constructs a MOO model based on these two indicators. Meanwhile, the study also uses EnergyPlus software. EnergyPlus software can ensure the accuracy of building models to a certain extent [17]. The multi-objective green building energy-saving optimization model is shown in equation (1).

$$\min F = \left(EC(X), DL(X) \right)$$
$$s.t. \quad X = \begin{pmatrix} x_{pd}, x_{epd}, x_{or}, x_{srar}, x_{wl}, x_{wh}, x_{ghtc}, \\ x_{shgc}, x_{hst}, x_{cst}, x_{tolw}, x_{lpd}, \end{pmatrix} \tag{1}$$

In equation (1), EC represents building energy consumption. DL represents the user's discomfort level. x_{or} represents the orientation of the room. In the air conditioning system, x_{cst} represents the cooling setting temperature. x_{tolw} is the thickness of the external insulation layer of the wall. x_{hst} represents the heating setting temperature. x_{wl} represents the window length. x_{epd} is the device power density. x_{ghtc} represents the heat transfer coefficient. x_{wh} represents the window height. x_{lpd} is the lighting power density. x_{srar} represents the solar absorption rate of the exterior wall. x_{shgc} represents the solar heat acquisition rate of the window. x_{pd} is the personnel density. The Bare-Bones PSO (BBPSO) can solve single objective problems. The particle update method is shown in equation (2) [18-19].

$$x_{i,j}(t+1) = \mathrm{N}\left(\frac{pb_{i,j}(t) + gb_j(t)}{2}, \left| pb_{i,j}(t) - gb_j(t) \right| \right) \tag{2}$$

In equation (2), N represents the particle swarm size. i represents the i-th particle. $gb_j(t)$ stands for the global optimum of i-th particles. $pb_{i,j}(t)$ stands for the j-th dimension of the optimal value for the i-th individual in the t-th iteration. t represents the number of iterations. This particle update method can be modified. The modified particle update method is shown in equation (3).

$$x_{i,j}(t+1) = \begin{cases} \mathrm{N}\left(\frac{pb_{i,j}(t) + gb_j(t)}{2}, \left| pb_{i,j}(t) - gb_j(t) \right| \right), U(0,1) < 0.5 \\ pb_{i,j}(t), \quad otherwise \end{cases} \tag{3}$$

In equation (3), $U(0,1)$ stands for a random number in [0, 1]. Figure 1 displays the BBPSO.

From Figure 1, firstly, initialize the population of the BBPSO. Next, update the particle position. Thirdly, determine whether the particle has crossed the boundary. If it is determined to be yes, the particles will be reinitialized. Otherwise, proceed to the next step. Fourthly, calculate the particle fitness. Fifthly, update the individual best and global best. Sixthly, determine whether the termination conditions are met. If it is met, the optimal solution is output and the process ends. Otherwise, the number of iterations is increased by 1 before returning to the second step. To apply the BBPSO to solving MOO problems, a Gaussian distribution based on the global and individual best of particles is applied to update particle positions. A Bare-Bones MOPSO (BBMOPSO) with few control parameters is formed. The particle update method of BBMOPSO algorithm is shown in equation (4).

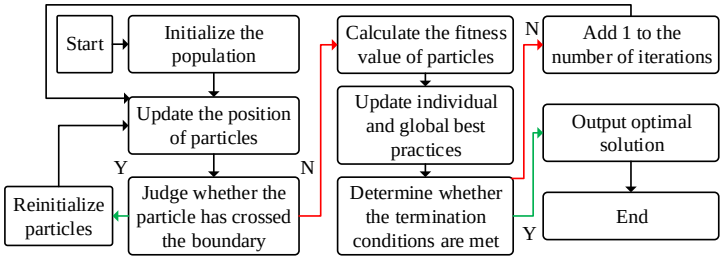


Fig.1 The basic process of BBPSO

$$x_{i,j}(t+1)=\begin{cases}\mathrm{N}\left(\dfrac{r_3\times pb_{i,j}(t)+(1-r_3)\times gb_{i,j}(t)}{2},\left|pb_{i,j}(t)-gb_{i,j}(t)\right|\right), if\ U(0,1)<0.5\\ pb_{i,j}(t),\quad otherwise\end{cases} \tag{4}$$

$$x_{i,j}(t+1)=\begin{cases}\mathrm{N}\left(\dfrac{r_3\times pb_{i,j}(t)+(1-r_3)\times gb_{i,j}(t)}{2},\left|pb_{i,j}(t)-gb_{i,j}(t)\right|+\delta_D\right), if\ U(0,1)<0.5\\ pb_{i,j}(t),\quad otherwise\end{cases} \tag{5}$$

In equation (4), r_3 represents a random number within the range $[0,1]$. $gb_{i,j}(t)$ stands for the j-th dimension of the i-th global optimal point in the t-th iteration. However, the BBMOPSO algorithm still suffers from the particles repeatedly searching for known regions and wasting computational resources. Therefore, the BBMOPSO algorithm is improved by introducing adaptive disturbance factors, resulting in the final BBMOPSO-A algorithm. The particle update method after introducing the adaptive disturbance factor is shown in equation (5).

In equation (5), δ_D represents the disturbance factor, as shown in equation (6).

$$\delta_D=\begin{cases}\left(x_D^{up}-x_D^{low}\right)\times e^{\left(-5t/T\right)}, pro\_d\geq rand\\ 0,\quad otherwise\end{cases} \tag{6}$$

In equation (6), T stands for the maximum iterations. x_D^{up} stands for the upper bound of the D-th decision variable value. x_D^{low} stands for the lower bound for the variable value. pro_d represents the disturbance probability. The specific calculation is shown in equation (7) [20].

$$pro\_d=0.5\times\left(1-\frac{1}{M}\sum_{m=1}^{M}\left|\frac{f_m(Pb_i(t))-f_m(Gb_i(t))}{f_m^{\max}-f_m^{\min}}\right|\right) \tag{7}$$

In equation (7), M is the total objective function. m represents the m-th objective function. $f_m^{\max}$ and $f_m^{\min}$ stand for the maximum and minimum values of all solutions in the reserve set regarding the m-th objective function, respectively. $f_m(Pb_i(t))$ and $f_m(Gb_i(t))$ represent the m-th objective function values of Gb_i and Pb_i, respectively. Figure 2 displays the BBMOPSO-A.

From Figure 2, firstly, set the the required parameters of BBMOPSO-A. Secondly, generate the particle position and set the feasible reserve set to an empty set. Thirdly, calculate the objective function value. Fourthly, update the external reserve set. Fifthly, determine whether the maximum iterations are reached. If it is achieved, output the result. Otherwise, proceed to the next step. Sixthly, select the global extreme point of each particle. Seventhly, determine the individual extreme points of particles through the Pareto dominance relationship. Eighthly, generate new particle positions and perform consistent mutations, and then return to the third step. The research uses MATLAB to implement the designed algorithm. The specific process of BBMOPSO-A algorithm combined with MATLAB and EnergyPlus software is shown in Figure 3.

From Figure 3, firstly, build a room model of a green building in the EnergyPlus software. Secondly, initialize the variable parameters. Thirdly, obtain the results of the EnergyPlus software. Fourthly, use MATLAB to write the BBMOPSO-A algorithm and update the particle positions. Fifthly, input the decision variable value into EnergyPlus. Sixthly, start and read the results in EnergyPlus through MATLAB. Seventhly, determine whether the result meets the standard. If it meets the standard, the process can be ended. Otherwise, it returns to the fourth step.

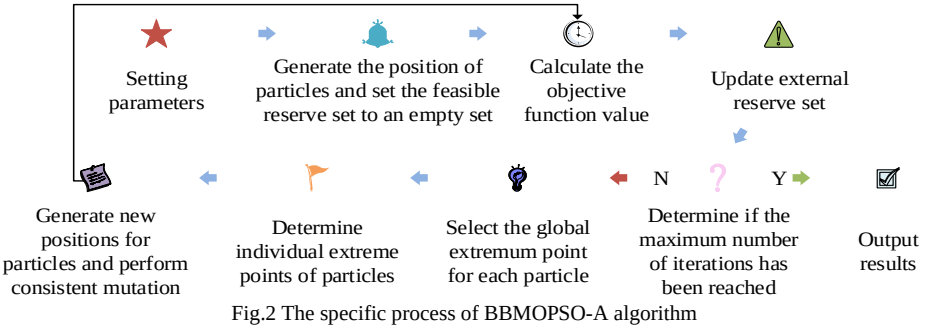


Fig.2 The specific process of BBMOPSO-A algorithm

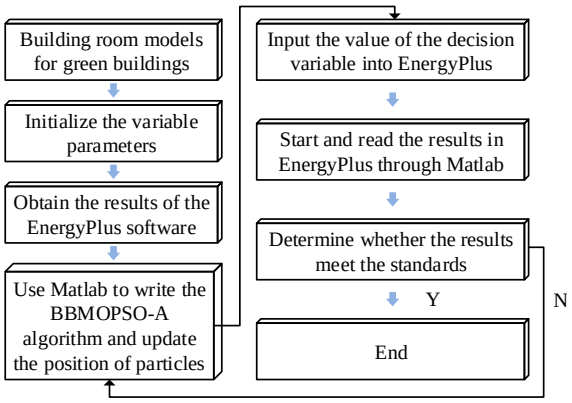


Fig.3 The specific process of BBMOPSO-A algorithm after integrating MATLAB and EnergyPlus software

B. *SMOPSO/D algorithm design for green building energy-saving design*

The BBMOPSO-A still has high running cost. To solve the computational time problem of BBMOPSO-A algorithm, a low computational cost proxy model assisted MOPSO is designed. This method first constructs the proxy model. Then the proposed Surrogate-model assisted Multi-Objective PSO based on Decomposition (SMOPSO/D) is designed. Figure 4 displays the SMOPSO/D.

From Figure 4, firstly, for the SMOPSO/D algorithm framework, predict new particles and their target values. Secondly, input the predicted target value into reserve set 1. Thirdly, obtain both global and individual guides. Fourthly, update the particles and other operators. Fifthly, transfer the new particles and solutions to the proxy model. Sixthly, determine whether to update the proxy model again. If the result is no, output the optimal solution set and end the process. Otherwise, proceed to the next step. Seventhly, implement a representative solution selection mechanism. Eighthly, truly evaluate the representative solution. The newly generated representative solution and the true target value are input into reserve set 2. Ninthly, retrain the training agent model. The tenth step is to output the Pareto optimal solution set and end the process. The study adopts a dual reserve sets collaborative guided variable sample size proxy model management strategy. The first step of this management strategy is to build an initial proxy model. The second step is to generate specific proxy model management strategies. When constructing the initial proxy model, Latin hyper-cubes are used for sampling. It serves as a sample for building the model. The number of samples required for the initial proxy model is shown in equation (8) [21].

$$Q=\frac{(n+1)(n+2)}{2} \tag{8}$$

In equation (8), n represents the number of randomly selected points. The proxy model update strategy with variable sample size guided by dual reserve sets collaboration is mainly divided into three parts. The first part is to determine the update timing of the proxy model. In response to the increased training cost of updating the proxy model, the average prediction error is used to determine the update timing. The average prediction error for reserve set 1 is shown in equation (9).

$$E(f,EP)=\frac{1}{\partial}\sum_{i=1}^{\partial}\sum_{q=1}^{\partial}\frac{\left|\hat{f}_q(x_i^{'})-f_q(x_i^{'})\right|}{\left|f_q(x_i^{'})\right|} \tag{9}$$

In equation (9), $\hat{f}_q$ represents the q-th objective function value obtained by fitting the proxy model. f_q represents the true objective function value calculated by the energy consumption software. EP is the set of endpoints selected from reserve set 1. ∂ is the number of endpoints. $x_i^{'}$ is the decision variable. The second part is the new sample selection strategy. To evaluate the overall similarity between the two datasets, the Hausdorff distance is adopted, as shown in equation (10) [22].

$$H(A,B)=\max\left(h(A,B),h(B,A)\right) \tag{10}$$

In equation (10), A and B represent two different datasets, respectively. $h(A,B)$ represents the directed Hausdorff distance from datasets A to B. $h(B,A)$ stands for the reverse Hausdorff distance. The $h(A,B)$ is shown in equation (11).

$$h(A,B)=\max_{a_\varphi\in A}\min_{b_\gamma\in B}\left|a_\varphi-b_\gamma\right| \tag{11}$$

In equation (11), $\left|a_\varphi-b_\gamma\right|$ stands for the distance norm between a and b. a_φ and b_γ are points in datasets A and B, respectively. The $h(B,A)$ is shown in equation (12).

$$h(B,A)=\max_{a_\varphi\in B}\min_{b_\gamma\in A}\left|a_\varphi-b_\gamma\right| \tag{12}$$

The specific steps of the new sample selection strategy guided by the dual reserve sets are shown in Figure 5.

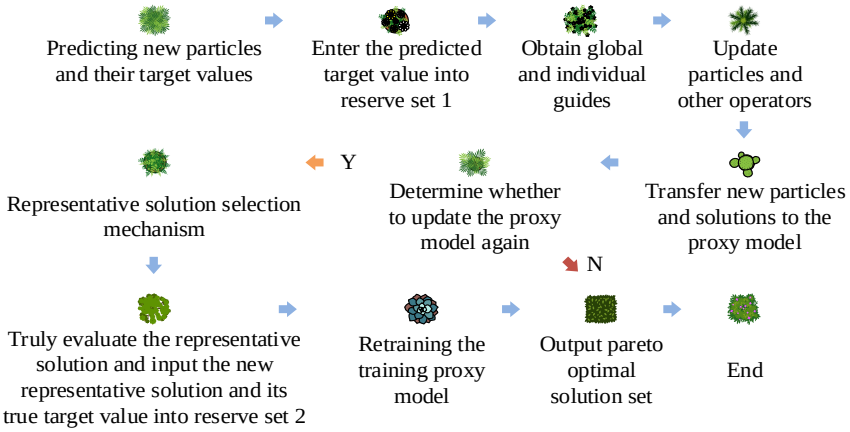


Fig.4 The specific framework of SMOPSO/D algorithm

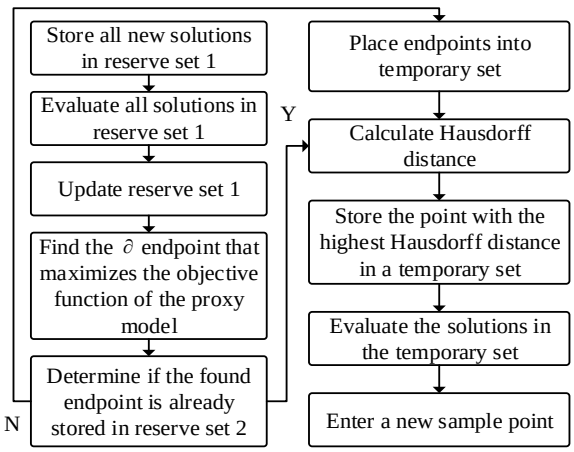


Fig.5 Specific steps of a new sample selection strategy guided by dual reserve sets

From Figure 5, the first step of the new sample selection strategy is to store all new solutions in reserve set 1. The second step is to evaluate all solutions in reserve set 1. The third step is to update reserve set 1. The fourth step is to find the ∂ endpoints in reserve set 1 that optimize the objective function. The fifth step is to determine whether the found endpoint has been stored in reserve set 2. If it is determined to be yes, proceed to the next step. Otherwise, the endpoint is placed in the temporary collection before proceeding to the next step. The sixth step is to calculate the non inferior points in reserve set 1 and endpoint set, as well as the Hausdorff distance of the elements in reserve set 2. The seventh step is to store the point with the maximum Hausdorff distance in a temporary set. The eighth step is to evaluate the solutions in the temporary set. The ninth step is to input a new sample point. The third part of the proxy model update strategy is to adaptively adjust the size of newly added samples. The new sample size is shown in equation (13).

$$L = l_{\min} + \left\lceil E\left[f, EP\right] \times \left(l_{\max} - l_{\min}\right) \right\rceil \tag{13}$$

In equation (13), $l_{\min}$ represents the minimum sample size. $l_{\max}$ represents the maximum sample size. $E\left[f, EP\right]$ is the model prediction error. $\lceil \bullet \rceil$ represents an upward rounding function. The SMOPSO/D algorithm utilizes the decomposition idea of the Multi-objective Optimization Evolutionary Algorithm based on Decomposition (MOEA/D). To generate initial particles with good distribution, a population initialization strategy based on crowding degree is introduced in the study. To ensure the diversity of the particle search method, the particle global and local guide update strategies based on decomposition are also introduced. The optimal solution of the aggregation function is shown in equation (14) [23].

$$\min g\left(x \mid \lambda_\varepsilon, z^*\right) = \max_{1 \le q \le R} \left\{ \lambda_\varepsilon^q \mid \hat{f}_q\left(x\right) - z_q^* \right\} \tag{14}$$

In equation (14), λ_ε represents the ε-th weight vector. R is the number of weight vectors. z^* is the objective function. z_q^* represents the objective function. λ_ε^q is the ε-th weight vector of the proxy model. The objective function z^* is shown in equation (15).

$$z^* = \min\left\{ \hat{f}_q\left(x\right) \mid x \in \Omega, q \in \left\{1, 2, ..., R\right\} \right\} \tag{15}$$

In equation (15), Ω represents the range of the independent variable x. The SMOPSO/D algorithm is displayed in Figure 6.

From Figure 6, the first step of the SMOPSO/D algorithm is initialization. The second step is to calculate the predicted the particle's objective function value. The third step is to store the new solution in reserve set 1. The fourth step is to update reserve set 1. The fifth step is to determine whether to update the proxy model. If updates are necessary, the size of the newly added samples is calculated. Otherwise, all optimal endpoints are stored in reserve set 2. The sixth step is to update the individual and global extreme points of the particles. The seventh is to update the particle position. Finally, whether the termination conditions are met is determined. If it is satisfied, the Pareto optimal solution is output and the process is ended. Otherwise, it will be returned to the second step.

IV. Analysis of Green Building Energy-Saving Design Based on MOPSO

In this chapter, the performance of the BBMOPSO-A and the SMOPSO/D algorithm is verified. Comparison algorithms and indicators are selected for comparison. The experimental environment and parameters are set. The selected comparison indicators include hypervolume measure and SC measure [24].

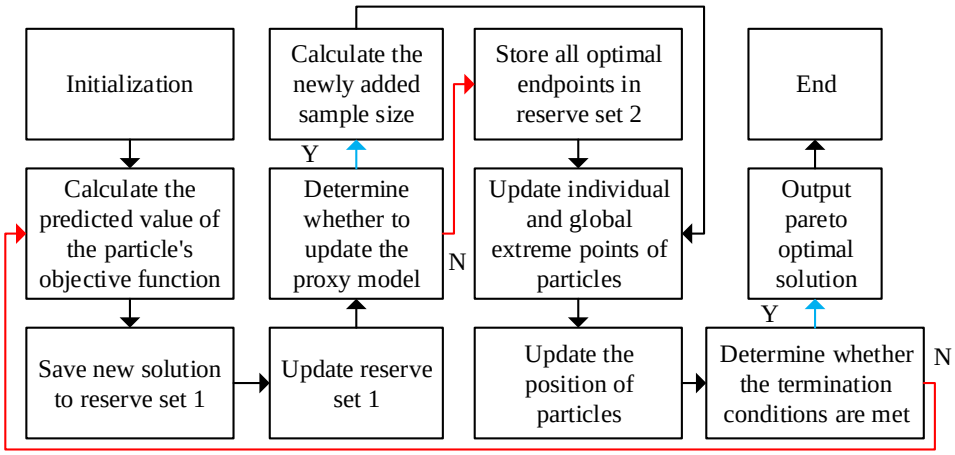


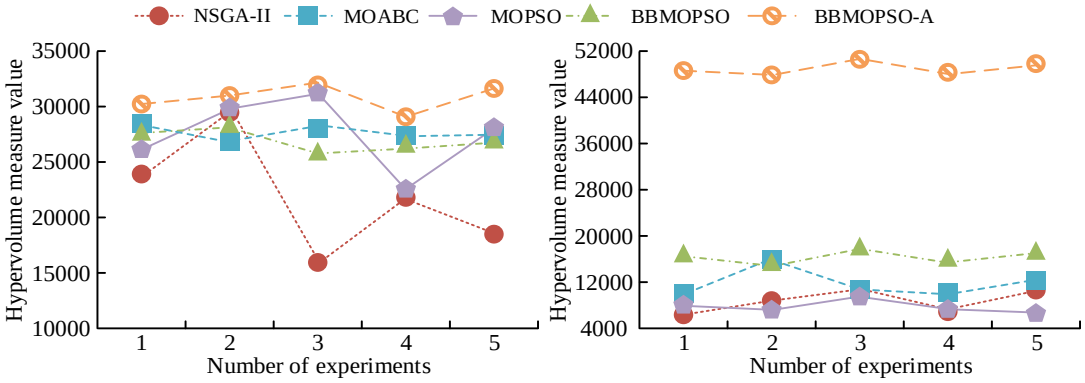
Fig.6 The specific process of SMOPSO/D algorithm

A. Result analysis of BBMOPSO-A for green building energy-saving design

To validate the BBMOPSO-A, three other methods are selected for comparison, namely Non-dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Artificial Bee Colony (MOABC), and MOPSO. The evaluation indicators are the hypervolume measurement and the SC measurement. The initial population size of different algorithms is 60. The maximum iterations are 25. The experiment selects energy-saving design of single room and three bedroom buildings in Tianjin as examples. EnergyPlus v9.5.0 software and MATLAB R2020a software are used. The central processing unit applied in the experiment is Intel Core i5-12600K. The operating system is Windows 10 with 16 threads. The memory is 128GB. A total of 5 experiments are conducted. The hypervolume measurements using different algorithms on single room and three bedroom buildings are shown in Figure 7.

From Figure 7 (a), in single room buildings, the maximum hypervolume measurement of the NSGA-II was 29765, the minimum was 15226, and the average was 21677. The maximum hypervolume measurement of the MOABC was 29040, the minimum was 27461, and the average was 27899. The maximum hypervolume measurement of the MOPSO hypervolume measurement was 30041, the minimum was 23194, and the average value was 27277. For the BBMOPSO algorithm, the maximum of the hypervolume measurement was 28489, the minimum was 25554, and the average was 27974. For the BBMOPSO-A algorithm, the maximum of the hypervolume measurement was 31264, the minimum was 28197, and the average was 29311. From Figure 7 (b), in the three bedroom building, for the NSGA-II algorithm, the maximum of the hypervolume measurement was 10632, the minimum was 4611, and the average was 6169.5. For the MOABC algorithm, the maximum of the hypervolume measurement was 12784, the minimum was 9087, and the average was 9251.9. For the MOPSO algorithm, the maximum of the hypervolume measurement was 9998, the minimum was 5808, and the average was 7075.3. For the BBMOPSO algorithm, the maximum of the hypervolume measurement was 15985, the minimum was 14097, and the average was 14322. For the BBMOPSO-A algorithm, the maximum of the hypervolume measurement was 50243, the minimum was 48907, and the average was 49504. The BBMOPSO-A algorithm performed better. The SC measurements in single room and three bedroom buildings are shown in Figure 8.

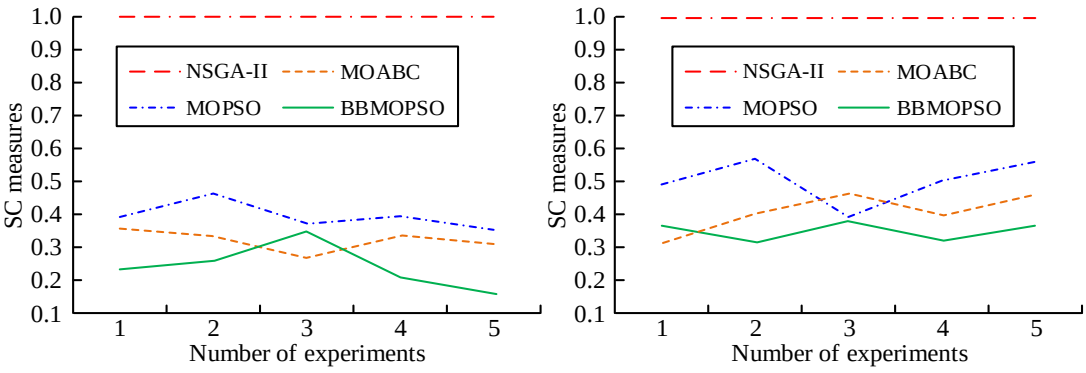
In Figure 8 (a), in a single room building, the maximum, minimum, and average SC measurements of the NSGA-II algorithm were all 1. For the MOABC algorithm, they were 0.3376, 0.2968, and 0.3172. For the MOPSO algorithm, they were 0.4611, 0.3367, and 0.3807. For the BBMOPSO, they were 0.3213, 0.1722, and 0.2246. From Figure 8 (b), in the three bedroom building, the maximum, minimum, and average SC measures of the NSGA-II algorithm were all 1. For the MOABC algorithm, they were 0.4312, 0.3127, and 0.3931. For the MOPSO, they were 0.5842, 0.3314, and 0.4546. For the BBMOPSO algorithm, they were 0.3817, 0.3284, and 0.3553. The Pareto frontiers in different algorithms in single room and three bedroom buildings were shown in Figure 9.



(a) Comparison of hypervolume measures on single room buildings

(b) Comparison of Hypervolume measures on three bedroom buildings

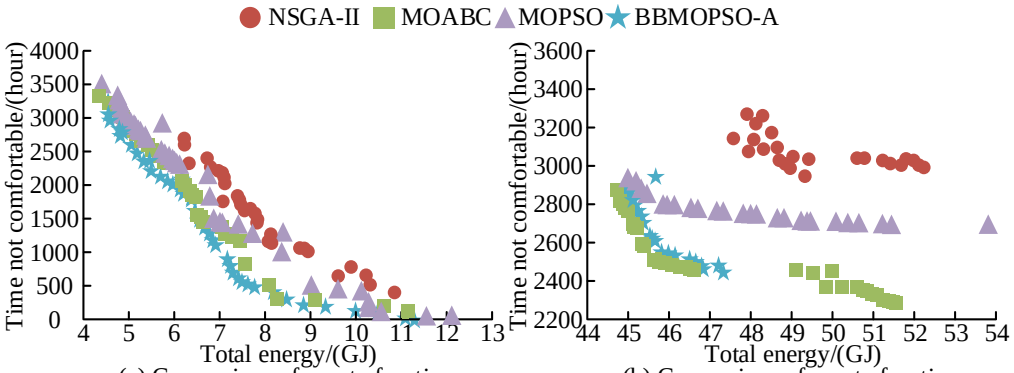
Fig.7 Comparison of hypervolume measurements using different algorithms in single room and three bedroom buildings



(a) Comparison of SC measures on single room buildings

(b) Comparison of SC measures on three bedroom buildings

Fig.8 Comparison of SC measures using different algorithms in single room and three bedroom buildings



(a) Comparison of pareto frontiers on single room buildings

(b) Comparison of pareto frontiers on three bedroom buildings

Fig.9 Comparison of Pareto frontiers of different algorithms in single room and three bedroom buildings

TABLE I
COMPARISON OF SOLVING TIME AND CPU UTILIZATION OF DIFFERENT ALGORITHMS

Algorithm	Time consuming/ms Number of experiments					CPU utilization/% Number of experiments				
	1	2	3	4	5	1	2	3	4	5
NSGA-II	378	391	412	437	409	25.98	26.77	27.51	28.94	28.46
MOABC	264	249	258	286	241	15.84	16.73	15.22	17.95	16.28
MOPSO	237	254	263	277	230	16.27	15.72	13.24	15.21	14.43
BBMOPSO-A	95	72	63	47	66	3.71	5.82	7.59	6.42	5.14

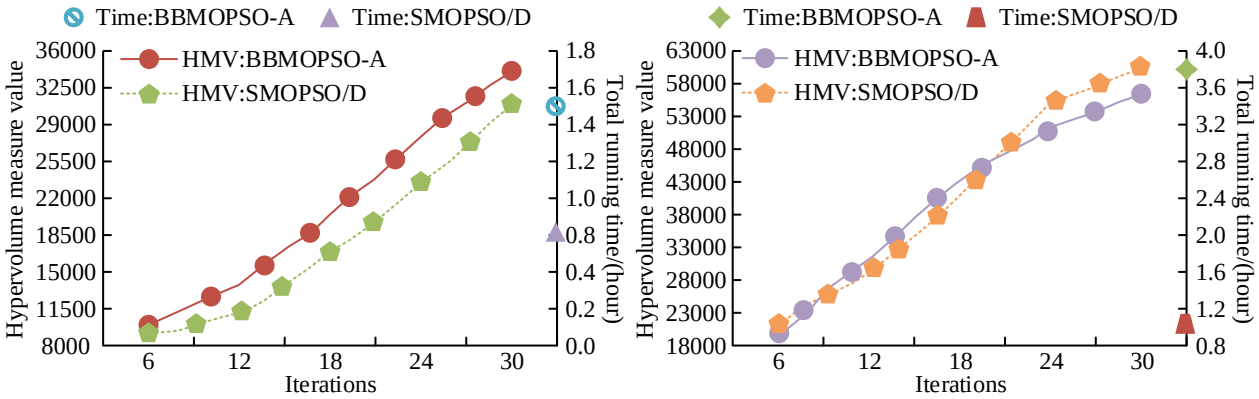
In Figure 9, the horizontal axis represents total energy consumption, and the vertical axis represents uncomfortable time. From Figure 9(a), in the single room building, the maximum uncomfortable time obtained by the NSGA-II algorithm was 2632 and the minimum value was 487. For the MOABC algorithm, they were 3314 and 13. For the MOPSO algorithm, they were 3448 and 0. The maximum uncomfortable time obtained by the BBMOPSO-A algorithm was 3601, and the minimum value was 0. In Figure 9(b), in the three bedroom building, the maximum uncomfortable time obtained by the NSGA-II algorithm was 3253, and the minimum value was 2947. For the MOABC algorithm, they were 2889 and 2283. For the MOPSO algorithm, they were 2900 and 2671. The maximum uncomfortable time obtained by the BBMOPSO-A algorithm was 2901, and the minimum value was 2450. From this, the BBMOPSO-A algorithm had better performance. The comparison of solution time and CPU utilization of different algorithms is shown in Table I.

From Table I, the average time consumption of BBMOPSO-A was 68.6ms, which was 336.8ms, 191.0ms, and 183.6ms lower than the average values of NSGA-II, MOABC, and MOPSO, respectively. This indicates that the BBMOPSO-A algorithm can solve multi-objective functions faster. In addition, the average CPU utilization of the four algorithms was 27.532%, 16.404%, 14.974%, and 5.736%, respectively. The BBMOPSO-A algorithm has lower CPU utilization when solving multi-objective functions. Overall, the BBMOPSO-A algorithm performs better.

B. Analysis of SMOPSO/D algorithm for green building energy-saving design

To verify the SMOPSO/D algorithm, the BBMOPSO-A algorithm is selected for comparison. The comparison indicators include the hypervolume measurement, SC measurement, Coefficient of Variation (CV) of root mean square deviation, and Mean Absolute Percent Error (MAPE). The number of iterations was 30. The software used in the experiment is consistent with the software used for performance verification of the BBMOPSO-A algorithm. The hypervolume measurements between BBMOPSO-A algorithm and SMOPSO/D algorithm in energy-saving design of single room and three bedroom buildings are shown in Figure 10.

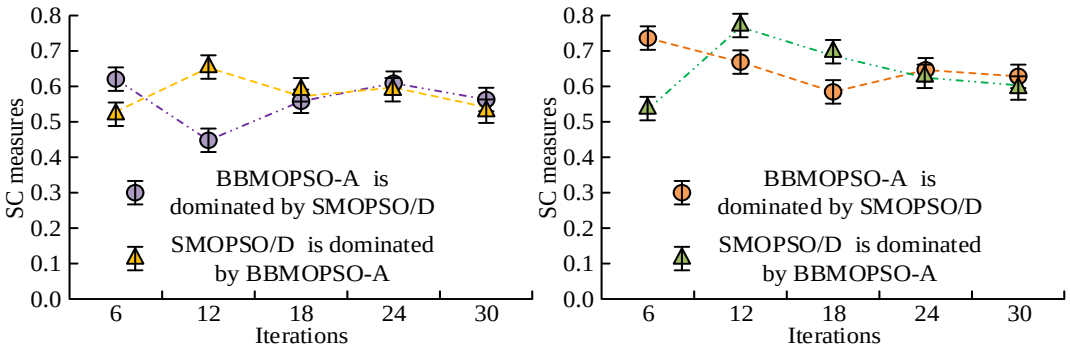
In Figure 10(a), in a single room building, as the maximum number of iterations increased, the hypervolume measurement value also increased. For the hypervolume measurement of the BBMOPSO-A, the maximum, minimum, and average values were 34726, 9171, and 24892. For the SMOPSO/D, the maximum, minimum, and average values were 33491, 8611 and 21153. In addition, the total running time of the BBMOPSO-A was 1.5 hours. The SMOPSO/D was 0.8 hours. From 10 (b), in the three bedroom building, for the BBMOPSO-A algorithm, the maximum of the hypervolume measurement was 57096, the minimum was 19191, and the average value was 40230. For the SMOPSO/D algorithm, the maximum, minimum and average values were 58624, 20564, and 40092. In addition, the total running time of the BBMOPSO-A was 3.8 hours. The SMOPSO/D was 1.1 hours. The performance of SMOPSO/D algorithm was superior to BBMOPSO-A algorithm. The SC measurements between BBMOPSO-A algorithm and SMOPSO/D algorithm in energy-saving design of single room and three bedroom buildings are shown in Figure 11.



(a) Comparison of Hypervolume measures on single room buildings

(b) Comparison of Hypervolume measures on three bedroom buildings

Fig.10 Comparison of overvolume measurement between BBMOPSO-A algorithm and SMOPSO/D algorithm in energy efficiency design of single-room and three-room buildings



(a) Comparison of SC measures on single room buildings

(b) Comparison of SC measures on three bedroom buildings

Fig.11 Comparison of SC measures between BBMOPSO-A algorithm and SMOPSO/D algorithm in energy-saving design of single room and three bedroom buildings

From Figure 11 (a), at different maximum iterations, the BBMOPSO-A was dominated by the SMOPSO/D algorithm at a proportion of 0.617, 0.463, 0.527, 0.598, and 0.539, respectively. The proportion of SMOPSO/D algorithm being dominated by BBMOPSO-A algorithm was 0.533, 0.654, 0.554, 0.557, and 0.536, respectively. From Figure 11 (b), at different maximum iterations, the proportion of BBMOPSO-A algorithm being dominated by SMOPSO/D algorithm was 0.709, 0.655, 0.594, 0.646, and 0.639, respectively. The proportion of SMOPSO/D algorithm being dominated by BBMOPSO-A algorithm was 0.514, 0.723, 0.664, 0.631, and 0.623, respectively. From this, the SMOPSO/D has better performance. The CV and MAPE values between BBMOPSO-A algorithm and SMOPSO/D algorithm in energy-saving design of single room and three bedroom buildings are shown in Table II.

TABLE II
COMPARISON OF CV AND MAPE VALUES BETWEEN BBMOPSO-A ALGORITHM AND SMOPSO/D ALGORITHM IN ENERGY-SAVING DESIGN OF SINGLE ROOM AND THREE BEDROOM BUILDINGS

Algorithm	Situation	CV Number of experiments			MAPE Number of experiments		
		1	2	3	1	2	3
BBMOPSO-A	Energy consumption objective function	0.017	0.019	0.016	0.013	0.017	0.009

TABLE III
COMPARISON OF GD AND IGD WITH DIFFERENT ALGORITHMS

Algorithm	GD Number of experiments					IGD Number of experiments				
	1	2	3	4	5	1	2	3	4	5
PSO	0.71	0.62	0.65	0.70	0.71	7.44	6.23	7.32	6.82	7.95
GA	0.66	0.58	0.62	0.63	0.53	5.03	5.67	4.70	4.35	4.33
NSGA-II	0.47	0.53	0.42	0.48	0.54	3.46	3.32	4.09	3.84	3.88
LS-AW-PSO	0.41	0.37	0.35	0.33	0.36	2.49	2.77	2.44	3.05	2.90
BBMOPSO-A	0.23	0.18	0.15	0.17	0.20	1.11	1.10	1.30	1.39	1.72
SMOPSO/D	0.17	0.20	0.13	0.11	0.16	1.07	0.84	0.53	0.64	1.54

From Table II, the maximum CV of the BBMOPSO-A algorithm was 0.019 and the minimum CV was 0.016 on the energy consumption objective function. For the MAPE, they were 0.017 and 0.013. The maximum of CV for the SMOPSO/D algorithm was 0.014, and the minimum was 0.011. For the MAPE, they were 0.009 and 0.006. In terms of uncomfortable hours, the maximum CV value of BBMOPSO-A algorithm was 0.026 and the minimum was 0.022. The maximum of MAPE was 0.031 and the minimum was 0.024. The maximum of CV for the SMOPSO/D algorithm was 0.021, and the minimum was 0.019. The maximum of MAPE was 0.013 and the minimum was 0.009. From this, the SMOPSO/D algorithm had better performance and higher prediction accuracy. In order to further demonstrate the performance of the SMOPSO/D algorithm, other algorithms are selected for comparative verification in the study. The comparison algorithms include PSO, Genetic Algorithm (GA), NSGA-II, and PSO combining Least Square (LS) and Adaptive Weight PSO (LS-AW-PSO). The comparison of the Generic Distance (GD) and Inverted Generic Distance (IGD) of different algorithms is shown in Table III.

From Table III, the average values of the six algorithms on GD were 0.678, 0.604, 0.488, 0.364, 0.186, and 0.154, respectively. The SMOPSO/D algorithm had the smallest average GD value. This indicates that the SMOPSO/D algorithm had better convergence, and the approximate Pareto front solved by the algorithm was closer to the true Pareto front. Meanwhile, as for IGD, the average value of SMOPSO/D algorithm was 0.924, which was 6.228, 3.892, 2.794, 1.806, and 0.400 lower than the average values of PSO, GA, NSGA-II, LS-AW-PSO, and BBMOPSO-A, respectively. This indicates that the comprehensive performance of SMOPSO/D algorithm includes better convergence and distribution performance, and strong robustness.

V. Conclusion

To optimize the green building design, an improved BBMOPSO-A and a decomposition-based proxy model assisted MOPSO were designed. In single room and three bedroom buildings, the average values of hypervolume measurements for the NSGA-II algorithm were 21677 and 6169.5, respectively. For the MOABC algorithm, the average hypervolume measurements were 27899 and 9251.9, respectively. For the MOPSO algorithm, the average hypervolume measurements were 27277 and 7075.3, respectively. For the BBMOPSO algorithm, the average hypervolume measurements were 27974 and 14322, respectively. For the BBMOPSO-A algorithm, the average hypervolume measurements were 29311 and 49504, respectively. The hypervolume measurements of the BBMOPSO-A in single room and three bedroom buildings were significantly higher than those of the comparison algorithms. In single room buildings, the average ratios of NSGA-II algorithm, MOABC algorithm, MOPSO algorithm, and BBMOPSO algorithm dominated by BBMOPSO-A were 1, 0.3172, 0.3807, and 0.2246, respectively. In three bedroom buildings, the average ratios of NSGA-II algorithm, MOABC algorithm, MOPSO algorithm, and BBMOPSO algorithm dominated by BBMOPSO-A algorithm were 1, 0.3931, 0.4546, and 0.3553, respectively. The BBMOPSO-A algorithm had superior performance. In single room and three bedroom buildings, the total running time of the BBMOPSO-A algorithm was 1.5 hours and 3.8 hours, respectively. The total running time of the SMOPSO/D algorithm was 0.8 hours and 1.1 hours, respectively. The SMOPSO/D algorithm has less runtime and better performance. There are also certain shortcomings in the research. The optimization model for green building energy-saving design is relatively simple. Future research can incorporate more modules into the model.

References

[1] Z. Serat, S. A. Z. Fatemi, and S. Shirzad, "Design and Economic Analysis of On-Grid Solar Rooftop PV System Using PVsyst Software," *Archives of Advanced Engineering Science*, vol. 1, no. 1, pp. 63–76, 2023.

[2] B. Lin, H. Chen, Y. Liu, Q. He, and Z. Li, "A Preference-Based Multi-Objective Building Performance Optimization Method for Early Design Stage," *Building Simulation*, vol. 14, no. 3, pp. 477–494, 2021.

[3] P. S. Badal and R. Sinha, "A Multi-Objective Performance-Based Seismic Design Framework for Building Typologies," *Earthquake Engineering & Structural Dynamics*, vol. 51, no. 6, pp. 1343–1362, 2022.

[4] J. S. Pan, N. Liu, S. C. Chu, and T. Lai, "An Efficient Surrogate-Assisted Hybrid Optimization Algorithm for Expensive Optimization Problems," *Information Sciences*, vol. 561, no. 2, pp. 304–325, 2020.

[5] T. Janus and S. Engell, "Iterative Process Design with Surrogate-Assisted Global Flowsheet Optimization," *Chemie Ingenieur Technik*, vol. 93, no. 12, pp. 2019–2028, 2021.

[6] Y. Zhou, J. Cai, and Y. Xu, "Indoor Environmental Quality and Energy Use Evaluation of a Three-Star Green Office Building in China with Field Study," *Journal of Building Physics*, vol. 45, no. 2, pp. 209–235, 2021.

[7] L. Almeida, V. W. Y. Tam, K. N. Le, and Y. She, "Effects of Occupant Behaviour on Energy Performance in Buildings: A Green and Non-Green Building Comparison," *Engineering*, vol. 27, no. 8, pp. 1939–1962, 2020.

[8] H. Yue and X. Jia, "Application Analysis of Green Building Materials in Urban Three-Dimensional Landscape Design," *International Journal of Nanotechnology*, vol. 19, no. 12, pp. 1117–1129, 2022.

[9] A. Andiyan, "Green Architectural Design Concepts at Mixed-Use Building Retail Office & FnB," *Solid State Technology*, vol. 64, no. 2, pp. 6183–6191, 2021.

[10] N. H. Trng and D. N. Dao, "New Hybrid Between NSGA-III with Multi-Objective Particle Swarm Optimization to Multi-Objective Robust Optimization Design for Powertrain Mount System of Electric Vehicles," *Advances in Mechanical Engineering*, vol. 12, no. 2, pp. 85–114, 2020.

[11] G. Xu, K. Luo, G. Jing, X. Yu, X. Ruan, and J. Song, "On Convergence Analysis of Multi-Objective Particle Swarm Optimization Algorithm," *European Journal of Operational Research*, vol. 286, no. 1, pp. 32–38, 2020.

[12] H. P. H. Anh and C. V. Kien, "Optimal Energy Management of Microgrid Using Advanced Multi-Objective Particle Swarm Optimization," *Engineering Computations*, vol. 37, no. 6, pp. 2085–2110, 2020.

[13] C. Zhi, L. Yiliang, H. Ke, and X. Kai, "Optimal Design of a Nuclear Power Plant Condenser Control System Based on Multi-Objective Optimization Algorithm," *Nuclear Technology & Radiation Protection*, vol. 35, no. 2, pp. 95–102, 2020.

[14] C. L. Tsai and G. Fredrickson, "Using Particle Swarm Optimization and Self-Consistent Field Theory to Discover Globally Stable Morphologies of Block Copolymers," *Macromolecules*, vol. 55, no. 12, pp. 5249–5262, 2022.

[15] I. Dagal, B. Akn, and E. Akboy, "Improved Salp Swarm Algorithm Based on Particle Swarm Optimization for Maximum Power Point Tracking of Optimal Photovoltaic Systems," *International Journal of Energy Research*, vol. 46, no. 7, pp. 8742–8759, 2022.

[16] X. Zhang, S. Zhang, "Low-Complexity Signal Detection for GSM Based on Group Detection," *Engineering Letters*, vol. 33, no. 4, pp. 1125-1134, 2025.

[17] I. B. Mansir, E. H. B. Hani, H. Ayed, and C. Diyoke, "Dynamic Simulation of Hydrogen-Based Zero Energy Buildings with Hydrogen Energy Storage for Various Climate Conditions,"

International Journal of Hydrogen Energy, vol. 47, no. 62, pp. 26501–26514, 2022.

[18] Z. Qiao, W. Shan, N. Jiang, A. A. Heidari, H. Chen, and Y. Teng, "Gaussian Bare-Bones Gradient-Based Optimization: Towards Mitigating the Performance Concerns," *International Journal of Intelligent Systems*, vol. 37, no. 6, pp. 3193–3254, 2022.

[19] W. Z. Sun, M. Zhang, J. S. Wang, S. S. Guo, M. Wang, and W. K. Hao, "Binary Particle Swarm Optimization Algorithm Based on Z-Shaped Probability Transfer Function to Solve 0-1 Knapsack Problem," *IAENG International Journal of Computer Science*, vol. 48, no. 2, pp. 294–303, 2021.

[20] J. Dai and L. Shu, "Fast-UAP: An Algorithm for Expediting Universal Adversarial Perturbation Generation Using the Orientations of Perturbation Vectors," *Neurocomputing*, vol. 422, no. 1, pp. 109–117, 2021.

[21] G. Chen, Y. Li, K. Zhang, X. Xue, and J. Yao, "Efficient Hierarchical Surrogate-Assisted Differential Evolution for High-Dimensional Expensive Optimization," *Information Sciences*, vol. 542, no. 2, pp. 228–246, 2020.

[22] W. Wu, X. Zhang, J. Zhang, Y. He, and W. Bai, "An Improved Hausdorff Distance Method for Locating Single Phase to Ground Fault in Neutral Non-Effectively Grounded System," *IET Generation, Transmission & Distribution*, vol. 15, no. 19, pp. 2747–2759, 2021.

[23] Y. Du, M. Liu, and L. Guo, "Scattering Phase Function of Fractal Aggregates of TiO2 Particulate Photocatalyst Simulated with Discrete Dipole Approximation," *International Journal of Hydrogen Energy*, vol. 45, no. 52, pp. 28034–28043, 2020.

[24] J. Q. Hale, H. Zhu, and E. Zhou, "Domination Measure: A New Metric for Solving Multiobjective Optimization," *INFORMS Journal on Computing*, vol. 32, no. 3, pp. 565–581, 2020.