Renewable Energy 134 (2019) 1295–1306

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Ineffectiveness of optimization algorithms in building energy optimization and possible causes

Binghui Si [a, b], Zhichao Tian [a, b], Xing Jin [a, b], Xin Zhou [a, b], Xing Shi [a, b, *]

[a] School of Architecture, Southeast University, Nanjing, China
[b] Key Laboratory of Urban and Architectural Heritage Conservation, Ministry of Education, China

A R T I C L E I N F O

Article history:
Received 2 February 2018
Received in revised form
21 July 2018
Accepted 17 September 2018
Available online 18 September 2018

Keywords:
Building energy optimization
Ineffectiveness of optimization algorithms
Reasons for ineffectiveness
Algorithm control parameters
Initial solution

A B S T R A C T

Building energy optimization (BEO) is an emerging technique for achieving energy-efficient building designs. The performance of optimization algorithms is crucial for achieving effective and efficient BEO techniques. In some cases, optimization algorithms can be ineffective, which results in the failure of the BEO process to identify an optimal design. Thus, it is important to investigate the reasons that cause algorithms to be ineffective in BEO. This study begins with a systematic definition of optimization algorithms' ineffectiveness, describing five ineffective scenarios. Then, a reference building and a representative energy optimization problem are proposed. Four commonly used optimization algorithms, namely, discrete Armijo gradient, Hooke-Jeeves, particle swarm optimization with constriction coefficient (PSOCC) and particle swarm optimization with inertia weight (PSOIW), are tested to determine the circumstances and the causal factors under which they become ineffective. The results shed more light on the performance of algorithms in BEO and can be used to help designers avoid ineffective algorithms.

1. Introduction

1.1. Background

The rapid growth of energy use worldwide has raised concerns about supply difficulties, the exhaustion of energy resources and severe environmental impacts (e.g., ozone layer depletion, global warming, and climate change). Specifically, the building sector is very energy intensive, accounting for approximately 39% of the primary energy use worldwide [1]. In this scenario, building designs that emphasize energy efficiency are therefore significant for achieving energy conservation and reducing environmental impacts.

As shown in several important review works [2,3], a new technique known as building energy optimization (BEO) has become a very active research field. Its benefits have been demonstrated to potentially reduce building energy use by as much as 30% resulting from a benchmark design [4]. As shown in Fig. 1, the BEO technique relies on optimization algorithms to generate new designs based on energy simulation results and predefined design objectives [5]. This technique has been widely used in the optimization of building envelopes (e.g., construction, form, double-skin facades), building systems (e.g., HVAC, lighting) and renewable energy generation (e.g., combined heat and power (CHP), solar technologies, ground energy and storage systems). For example, Lorestani and Ardehali [6] used a newly developed evolutionary particle swarm optimization (PSO) algorithm to develop a simulation model for optimization of an autonomous CHP system that incorporates renewable energy sources. Pereira and Aeleneia [7] used a genetic algorithm (GA) to optimize the efficiency of a building-integrated photovoltaic/thermal-phase change material (BIPV/T-PCM) installed in an office building façade. Baniassadi et al. [8] used a GA to optimize an off-grid wind turbine while considering the demand profile.

As shown in Fig. 1, optimization algorithms play a vital role in the BEO workflow. Therefore, the performance of optimization algorithms is significant for the effectiveness and efficiency of the BEO technique. The optimization algorithms commonly used in BEO are illustrated in Fig. 2. They can be grouped into three categories: direct search algorithms, intelligent optimization algorithms, and hybrid algorithms. According to our previous review [9], intelligent optimization algorithms are used most frequently in BEO, accounting for approximately 64% of the core literature, followed by direct search algorithms (approximately 16%) and hybrid

* Corresponding author. School of Architecture, Southeast University, 2 Si Pai Lou, Nanjing 210096, PR China.
E-mail address: shixing_seu@163.com (X. Shi).

https://doi.org/10.1016/j.renene.2018.09.057
0960-1481/

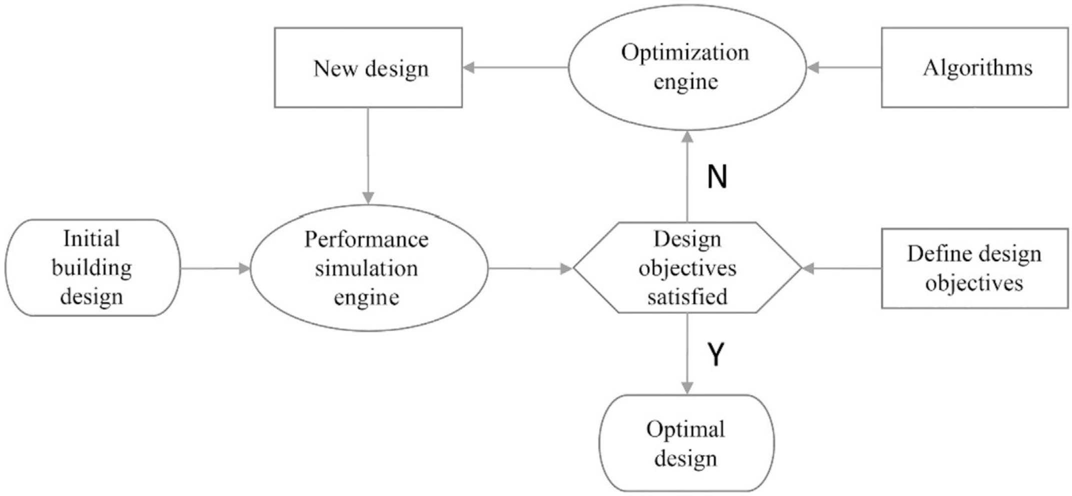

Fig. 1. Workflow of the BEO technique.

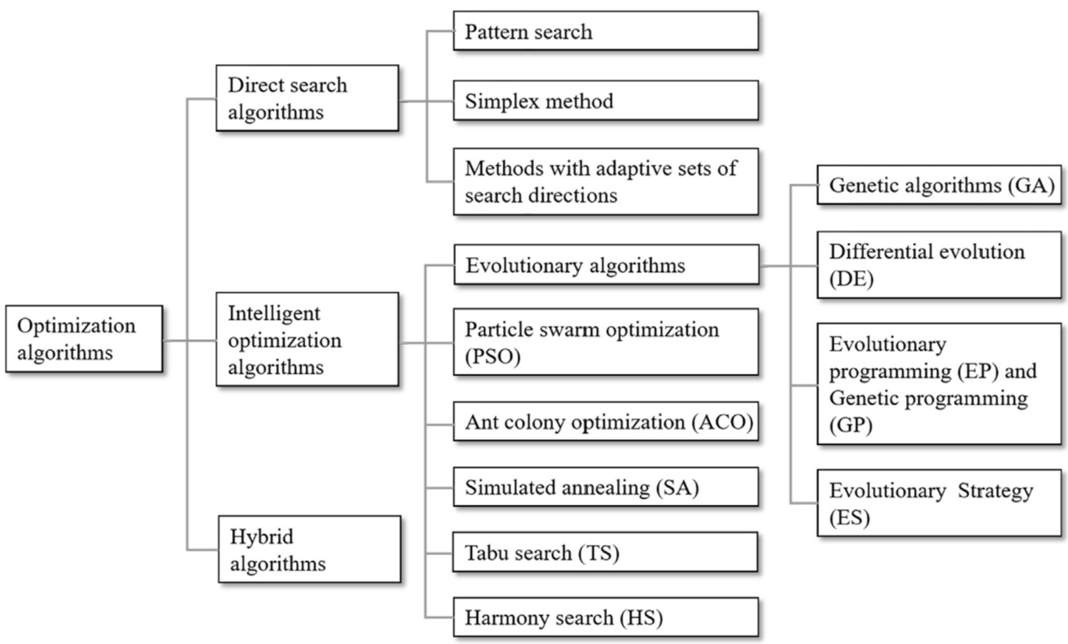

Fig. 2. Optimization algorithms commonly used in BEO.

algorithms (approximately 10%). Among the intelligent optimization algorithms, the GA and its modified versions dominate, accounting for approximately 41% of the core literature, followed by the PSO algorithm (approximately 13% of the core literature).

Although many algorithms are currently used in BEO, according to the so-called "no-free-lunch theorem," a general-purpose, universal optimization algorithm is, however, impossible [10]. It is difficult to find an optimization algorithm that performs well for all optimization problems. Similarly, not all algorithms are effective for any given building optimization problem. Therefore, in addition to focusing on the effective optimization algorithms, the ineffective optimization algorithms and the reasons for their ineffectiveness when they are used in BEO are equally worth studying. Such knowledge would allow designers to choose an appropriate algorithm among the available options and to help them avoid ineffective algorithms. However, existing works targeting the ineffectiveness of algorithms in BEO are rare. Moreover, some related problems have not yet been adequately addressed, such as a firm definition of algorithm ineffectiveness, the reasons that may cause an algorithm to be ineffective, and its risk of becoming ineffective.

1.2. Literature review

1.2.1. Studies focusing on the performance of optimization algorithms in nonarchitectural fields

The existing literature on the performance of optimization algorithms is mostly from nonarchitectural fields. However, a clear definition of ineffective optimization algorithms was not readily available in the literature. Most studies were conducted by comparing the performance of different algorithms based on a set of performance criteria to determine which algorithm is better than another. For example, different groups of test functions with difficult features have been proposed within the context of the IEEE

Congress on Evolutionary Computation (IEEE-CEC) competitions [11,12], which are of great importance for evaluating and comparing the performance of modified or newly proposed algorithms. Evaluation criteria, namely, success rate, convergence graphs, algorithm complexity, parameters, and encoding were proposed for the performance comparison of different algorithms. However, whether the performance evaluation results of optimization algorithms on test functions hold for BEO problems is still a question. One reason is the objective function evaluation of a test function is very fast when using MatLab or other mathematical software. However, the time needed to complete a simulation of a detailed building model varies from several minutes to several hours. In this case, an algorithm that is found to be effective for a test function may not achieve the same level of performance for a BEO problem because the maximum number of objective function evaluations available for BEO problems can be 100–1000 magnitudes smaller than that for test functions considering the limited computing time, which may greatly impact the quality of the final optimization results.

1.2.2. Studies focusing on the effectiveness and ineffectiveness of optimization algorithms in BEO

According to the recently published review in Ref. [9], only a few researchers in the field of building optimization have paid close attention to the performance (e.g., effectiveness or ineffectiveness) of algorithms used in BEO. Among them, the number of research studies focusing on ineffective algorithms and the reasons for their ineffectiveness are scarce. Most of the attention has been paid to the best algorithms. Those ineffective algorithms that behave poorly for optimization problems and the reasons that cause their poor behavior are usually ignored. For example, Hofpe [13] assessed the performance of two multiobjective optimization algorithms (MOOAs), the nondominated sorting genetic algorithm (NSGA-II) and the S-metric selection multiobjective evolutionary algorithm (SMS EMOA). She found that the performance of the NSGA-II was not satisfactory on all the tests, while the SMS EMOA yielded more competitive results but required a much higher number of simulations. However, the cause of the ineffectiveness of the NSGA-II remained unknown. Hamdy et al. [14] compared the effectiveness and efficiency of seven commonly used multiobjective evolutionary optimization algorithms in solving a design problem of a nearly zero-energy building. The results indicated that the two-phase optimization using the genetic algorithm (PR_GA) performed best, followed by the controlled nondominated sorting genetic algorithm with a passive archive (pNSGA-II), the multiobjective evolutionary algorithm based on the concept of epsilon dominance (evMOGA) and the multiobjective differential evolution algorithm (spMODE-II). In contrast, the elitist nondominated sorting evolution strategy (ENSES), the multiobjective particle swarm optimization (MOPSO) and the multiobjective dragonfly algorithm (MODA) algorithms achieved uncompetitive results in most cases. However, the reasons for those results were not explored. Therefore, this paper aims to fill the knowledge gaps regarding the ineffectiveness of optimization algorithms specifically used in BEO and help users avoid ineffective algorithms.

1.2.3. Studies that did not address algorithm performance in BEO

The vast majority of BEO studies simply apply some algorithms to specific optimization problems without paying attention to the performance of the algorithms, much less addressing the problems of ineffective algorithms or the reasons for their ineffectiveness. Sundareswaran and Palani [15] developed a new algorithm for maximum power point tracking (MPPT) in large PV systems under partial shading conditions (PSC). The new algorithm used the PSO algorithm for MPPT during the initial stages of tracking and then employed the traditional perturb and observe (PO) method at the final stages. Sharafi et al. [16] developed a simulation-based metaheuristic approach that determines the optimal size of a hybrid renewable energy system for residential buildings. A dynamic MOPSO algorithm was used to maximize the renewable energy ratio of buildings and minimize the total net present costs and CO_2 emissions for required system changes. Wang et al. [17] used a GA to optimize the configuration of a biomass gasification-based building, cooling, heating and power (BCHP) system with a thermal storage unit and hybrid cooling system to minimize the annual total cost (ATC).

1.3. Research outline

The research outline of this paper is as follows:

- Define the ineffectiveness of optimization algorithms in BEO.
- Present and explain typical situations where optimization algorithms are ineffective.
- Develop a representative BEO problem for algorithm performance evaluation.
- Evaluate four commonly used optimization algorithms and analyze the reasons why they become ineffective.

The reasons why the study presented in this paper is both valuable and timely are multifold. On the theoretical side, the performance of optimization algorithms is critical to the overall effectiveness and efficiency of BEO techniques. Although some algorithms are known to be ineffective for some BEO problems, the reasons for their ineffectiveness in these cases are unclear, and relevant research is scarce. Therefore, the results obtained from this study can deepen our understanding of optimization algorithms and of BEO in general. On the application side, understanding the reasons why optimization algorithms are ineffective can help architects, engineers, and consultants select the appropriate optimization algorithms and set their parameters to effectively apply the BEO technique and to better design and use various energy-saving and renewable energy generation measures.

The remainder of this paper is organized as follows: Section 2 proposes the definition of ineffective optimization algorithms in BEO. Section 3 presents five typical situations in which optimization algorithms become ineffective. For each situation, a clear description and a schematic diagram are given to assist the reader's understanding. Section 4 establishes a reference building and a representative optimization problem based on the medium office building developed by the U.S. Department of Energy (DOE). In Section 5, four commonly used optimization algorithms are selected and applied to the representative BEO problem. The reasons why they become ineffective are explored in detail. Finally, Section 6 concludes the paper.

2. Definition of optimization algorithms' ineffectiveness in BEO

For a given BEO problem, if an optimization algorithm can reliably find a satisfactory optimal solution within a given period of time, it is considered effective for that particular problem. It follows that an effective optimization algorithm should simultaneously meet three criteria: (1) be able to deliver a satisfactory optimal solution, (2) be able to complete the optimization process within a given time constraint, and (3) demonstrate good reliability. Therefore, an algorithm that violates any of these three criteria is considered ineffective. The three criteria are explained as follows.

2.1. Satisfactory optimal solution

For a BEO problem, the optimal solution obtained by an algorithm is deemed satisfactory if the desired solution quality is achieved. The quality of a solution can be measured by the difference between the optimal solution of the algorithm and the true optimum, as shown in Eq. (1).

$$\delta = \frac{|f(X') - f(X^*)|}{f(X^*)} \times 100\%, \quad (1)$$

where $f(X')$ is the objective function value of the optimal solution found by the algorithm, and $f(X^*)$ is the objective function value of the true optimum, which in some cases can be obtained through a brute-force search method. Note that sometimes, the true optimum of an optimization problem is unavailable; then, a reference solution can be used in the place of the true optimum, which can be obtained by running the optimization process as many times as possible and selecting the best solution.

Therefore, to determine whether the quality of an optimal solution is satisfactory, the value of δ calculated by Eq. (1) must be compared with a desired accuracy level δ_* which depends on how accurate the final optimization results need to be. When the value of δ is larger than that of δ_*, the quality of the optimal solution is considered unsatisfactory; thus, the algorithm is ineffective for that particular optimization process.

2.2. Time constraint

Due to limited time and resources, an optimization run must stop at some point. For an annual building energy simulation, the computation time required can vary from a few seconds [18] to several hours [19]. However, a BEO process typically involves hundreds or even thousands of annual building energy simulation runs to obtain a near-optimal solution. Therefore, the time required varies from several minutes to several hours or even days. To ensure that an optimization process is both timely and practical, the optimization algorithm must be able to find a satisfactory optimal solution within a limited time constraint. Otherwise, it is considered ineffective.

2.3. Good reliability

Good reliability implies that an optimization algorithm can consistently find a satisfactory optimal solution for a given optimization test when it is repeated multiple times with all relevant settings unchanged (e.g., selection of the initial solution, settings of algorithm control parameters, etc.). However, stochastic optimization algorithms (e.g., GA, PSO, etc.), which involve random operators in their optimization processes, can result in different optimal solutions after each repeated run. In this case, if such algorithms do not necessarily guarantee a satisfactory optimum for every repeated run, they can be ineffective for the given optimization problem. In addition, when a user is unsure which optimization run will achieve a satisfactory optimum, the same optimization test must be executed as many times as possible to select the best one, leading to extended computing time and considerable uncertainty.

Algorithm reliability can be measured using the success rate achieved by repeating the same optimization run multiple times. Specifically, a successful run converges to a satisfactory optimal solution within a limited time constraint. Eq. (2) provides a formula that, in essence, is the ratio of successful runs to the total runs.

$$\beta = \frac{N_{success}}{N_{total}} \times 100\%, \quad (2)$$

where $N_{success}$ is the number of successful runs, and N_{total} is the total number of repeated runs.

According to the low probability event (LPE) principle [20], which is an important theorem in probability and commonly applied in practical projects and mathematical statistics, an LPE is considered as one that will not occur in the actual environment. Generally, the probability of an LPE is considered as 1%, 5% or 10%. In this study, LPE values of 5% and 10% are used to rank the reliability level of an algorithm. Consequently, for a given BEO problem, when $100\% - \beta \le 5\%$, the reliability of the algorithm is perfect. When $5\% \le 100\% - \beta \le 10\%$, the algorithm's reliability is acceptable. However, when $100\% - \beta > 10\%$, the reliability of the algorithm is poor, and the algorithm is considered ineffective for the given optimization problem.

3. Typical situations in which optimization algorithms are ineffective in BEO

Five typical situations in which optimization algorithms are ineffective are presented and explained in detail in this section. In these situations, each algorithm violates at least one of the aforementioned three effectiveness criteria.

3.1. Falling into local optima traps

One potential situation in which an optimization algorithm is ineffective is when it becomes trapped in local optimum. In applied mathematics and computer science, a local optimum of an optimization problem is a solution that is optimal (either maximal or minimal) within a neighboring set of candidate solutions [21]. A local optimum differs from a global optimum in that the latter is the optimal solution among all possible solutions, not just those in a particular neighborhood of values.

The objective functions used in BEO problems, such as annual energy consumption or heating and cooling loads, are always multimodal. As Fig. 3 shows, the global (true) optimal solution is located at point O. Two local optimal solutions, points A and B, also exist in the figure. If an optimization algorithm can find only point B as the best solution, it is considered ineffective, because the objective function value at point B is significantly larger than that at point O, which indicates that the local optimal solution found is too far away from the global optimal solution. However, it should be noted that an optimization algorithm able to find point A as the best solution could be considered effective. Although point A is still a local optimal solution, it is so close to point O that the objective difference between the found local optimal solution and the global optimal solution is sufficiently small to be ignored.

In BEO, many factors may cause an algorithm to become trapped in a local optimum, including the algorithm's control parameter

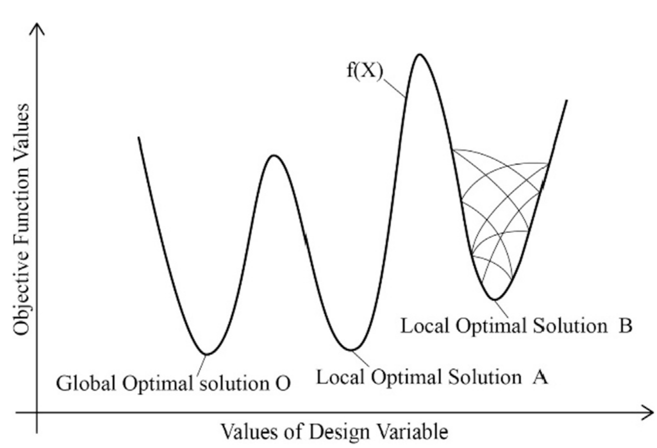

Fig. 3. A possible graph of a single-variable function.

settings, the position of the initial solution, the search step size, and the complexity of the optimization problem (e.g., the number of design variables). Therefore, it is important to use proper strategies to help optimization algorithms escape local optima.

3.2. Slow search speed

In BEO, search speed is vital for the effectiveness of an algorithm because the final optimization result can vary substantially across different runtime durations. When the runtime is restricted, high-speed optimization algorithms are more likely to achieve better solutions, and such algorithms can greatly reduce the computing time, especially for complex BEO problems in which cost function evaluation is usually time consuming.

Based on the aforementioned three effectiveness criteria, an effective algorithm should obtain a desirable optimal solution within the given time constraint. Therefore, when an algorithm searches so slowly that it consistently cannot find a satisfactory optimal solution before being terminated over successive runs, then it is considered ineffective for the given problem. This situation is further illustrated in Fig. 4, which shows an optimization run driven by an algorithm. The number of building simulations used by the optimization is restricted to 300. However, the algorithm's search is so slow that it is unable to find an optimal solution during the 300 simulations, and finally converges to the true optimum only at the 558th simulation, which far exceeds the time limit. Therefore, this algorithm is considered ineffective for this case.

In BEO, many factors can decelerate the search process of an algorithm and cause it to be ineffective. These factors include algorithm control parameter settings, the position of the initial solution, and the number of design variables. Hence, appropriate approaches and specific techniques should be used to significantly decrease the optimization runtime and help the algorithms reach an acceptable optimum as soon as possible.

3.3. Nonconvergence

In general, the ideal convergence process for an algorithm is to exhibit wide coverage and rapid global search during the initial stages with precise fine-tuning around a close-to-optimal solution in the later stages. It ensures that the optimization search is diversified and that the optimal solution found by the algorithm locates as closely as possible to the true optimum. However, as shown in Fig. 5, if an algorithm maintains wide global search coverage with large step sizes throughout the entire search process and fails to fine-tune around a near optimum, then the algorithm may be unable to converge to a solution with a quality as high as expected. Consequently, the algorithm can be ineffective in the optimization process.

In BEO, the factors that may cause algorithm convergence failures are multifarious. One possible factor is inappropriate algorithm parameter settings. Optimization algorithms usually have various parameters that must be defined to successfully execute the optimizing process, some of which control the convergence of algorithms. For example, when an evolutionary algorithm's crossover rate is large, its global search ability is strong, but its local search ability is weak, which may cause the algorithm to fail to precisely converge and eventually lead to its ineffectiveness.

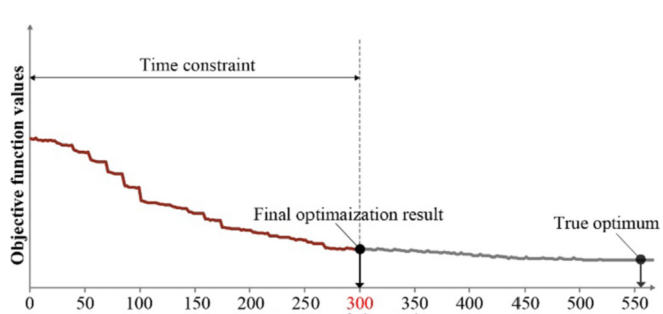

Fig. 4. An illustration of a failure optimization case with slow search speed.

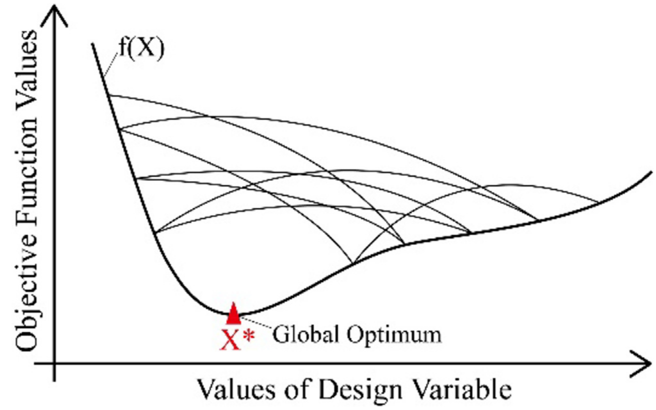

Fig. 5. An illustration of a nonconvergent optimization process driven by an algorithm.

3.4. Midway termination

As shown in Fig. 6, another possible situation in which an optimization algorithm can be ineffective is when the optimization run suddenly terminates somewhere before meeting any of the predetermined termination criteria. In this case, the algorithm is considered ineffective in the optimization run if no satisfactory optimal solution was found before termination.

In BEO, the reasons why an optimization run might terminate before completion are various and should be analyzed according to the specific optimization problem at hand. Such reasons include unexpected runtime errors, inappropriate algorithm control parameter settings, etc. For example, in BEO, the simulation output may become discontinuous at times when integer or discrete values are assigned to design variables. Even for optimization problems in which all inputs are continuous parameters, the nature of the algorithms in detailed building simulation programs itself often generates discontinuities in the simulation outputs [22]. This feature may cause some gradient-based optimization algorithms to be ineffective because they may terminate when encountering discontinuities.

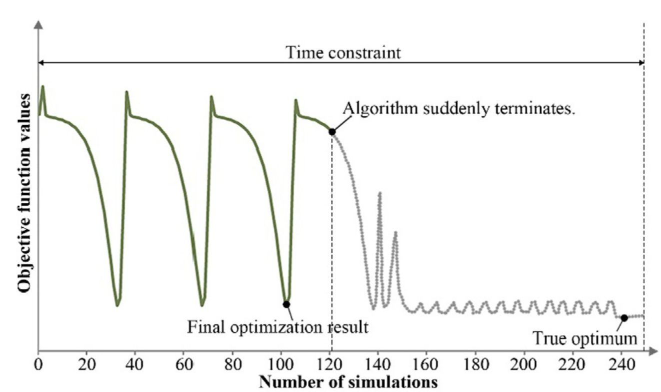

Fig. 6. An ineffective optimization case in which an algorithm suddenly terminates without converging.

3.5. Poor reliability

An effective algorithm should have good reliability, as explained in Section 2.3. Fig. 7 illustrates three different cases of reliability. Each point with the same shape represents the optimal solution obtained in each repeated run by the same algorithm. Clearly, all the triangle solutions are concentrated within the satisfactory solution scope. However, the circular solutions vary widely, and few are located in the satisfactory solution scope. Moreover, although the cross solutions are very similar to each other, which means the algorithm stably found similar solutions, none of them are satisfactory. Therefore, the reliability of algorithms 1 and 2 is poor, and they are considered ineffective for the given optimization problem. In contrast, Algorithm 3 can consistently find a satisfactory solution; therefore, its reliability is perfect in this case.

In BEO, many factors may cause an algorithm to lose reliability, for example, the algorithm's control parameter settings and the position of the initial solution in the design space.

4. Case study presentation

The previous sections defined the ineffectiveness of optimization algorithms and presented typical ineffective situations. In this section, four commonly used optimization algorithms are analyzed with respect to their ineffectiveness for the underlying causes. A reference building and a representative optimization problem are proposed, to which the four selected algorithms are applied.

4.1. Reference building

A 3-story rectangular office building following the model of the DOE medium office building [23] is used as the reference building in this study. Its total floor area is 4982 m^2, and the floor height is 4 m. Fig. 8 shows a perspective view of the building. As shown in Fig. 9, there is one core thermal zone and four perimeter thermal zones on each floor. The reference building is assumed to be located in Baltimore, Maryland. The wall and roof construction types are steel frame and IEAD, respectively, which were determined from an analysis of the 2003 Commercial Buildings Energy Consumption Survey (CEBCS) data [24] and other information by Pacific Northwest National Laboratory (PNNL) [25]. Note that the wall, roof, and window thermal parameters were set to the standard 90.1–2004 [26] values. Specifically, the opaque wall and roof thermal resistance are 1.42 m^2 K/W and 2.79 m^2 K/W, respectively. The window thermal properties include the following: the U-value is 3.24 W/$m^2 \cdot$K, the SHGC is 0.39, and the visible transmittance is 0.31. The building is equipped with a variable air volume HVAC system with a gas furnace and electric reheat. The sources of internal heat gains include electric equipment (10.76 W/m^2), lights (10.76 W/m^2), and heat release by people (18.58 m^2/person).

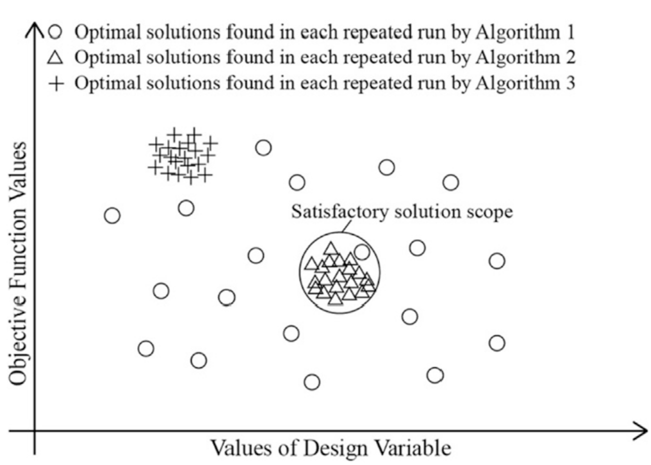

Fig. 7. An illustration of three different cases of reliability.

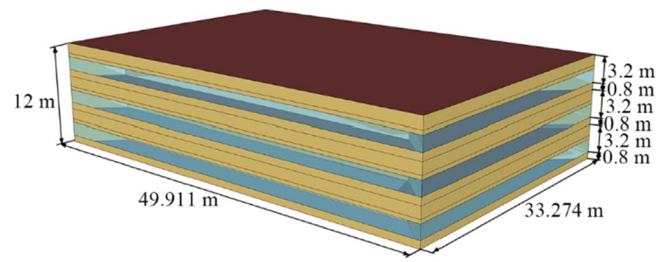

Fig. 8. A perspective view of the reference building.

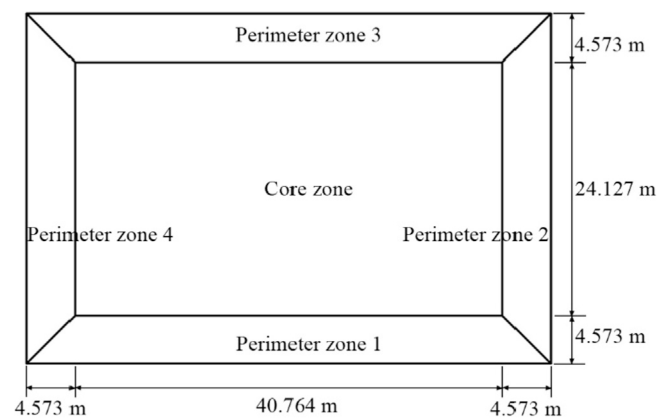

Fig. 9. Division of the thermal zones on each floor.

4.2. Representative optimization problem

In the present study, the optimization objective of all optimization runs is to minimize the annual energy consumption of the reference building as calculated by EnergyPlus [27]. In addition, the representative optimization problem consists of 7 continuous variables: the conductivity of the exterior wall insulation, the building orientation, the window upper positions in each façade, and the cooling design supply air temperature used for sizing the HVAC system. Specifically, the lower positions for all windows are fixed at 0.8 m from the floor, and the windows in the same façade have equal area. Table 1 lists the independent variables and their initial value, best value, step size, and range of variation.

4.3. Selected optimization algorithms

The four optimization algorithms analyzed in this paper are the discrete Armijo gradient algorithm [28], the Hooke-Jeeves algorithm [29] and two versions of PSO algorithms [30] (i.e., the particle swarm optimization with constriction coefficient (PSOCC) algorithm and the particle swarm optimization with inertia weight (PSOIW) algorithm). They belong to the "gradient-based," "pattern search" and "metaheuristic" classes of optimization algorithms, respectively. A comprehensive description of each algorithm is not provided in this paper, but interested readers are referred to [31] for more technical information and a detailed description of each algorithm.

Table 1
Specifications of optimization variables.

Design variables	Symbol	Unit	Step size	Range	True optimum	Initial value
Insulation Conductivity	x_1	W/m·K	0.0025	[0.01, 0.1]	0.01	0.049
Orientation	x_2	°	5	[30, 180]	180	120
South window upper position	x_3	m	0.05	[1.35, 2.7]	1.35	1.63
East window upper position	x_4	m	0.05	[1.35, 2.7]	1.35	1.8
North window upper position	x_5	m	0.05	[1.35, 2.7]	1.35	2.66
West window upper position	x_6	m	0.05	[1.35, 2.7]	1.35	1.35
Cooling supply air temperature	x_7	°C	0.2	[10, 20]	12.3	13

5. Numerical experiments and results

The performance of an optimization algorithm depends to some extent on its control parameters [32] and the initial solution. Therefore, it is natural to investigate how the algorithm control parameters and the position of the initial solution in the design space lead to the ineffectiveness of the four selected optimization algorithms.

In this study, GenOpt 3.1.1 [33] and EnergyPlus 8.3.0 [34] are used to perform all optimization tests. Each optimization test is repeated 10 times using each algorithm to evaluate the reliability of each. For each optimization run, the process is terminated when the number of simulations reaches 300. These numbers were chosen to strike a balance between what is preferred and what is practical in terms of computing time. All simulations were conducted on a computer with an Intel (R) Core (TM) i7-6700HQ CPU @ 2.60 GHz, 8 GB of main memory, and the Windows 10 operating system. Each simulation took approximately 30 s to complete. Thus, an optimization run with 300 simulations needed approximately 2–2.5 h, which is considered an appropriate duration of computing time.

5.1. Identifying the true optimum

To compare the quality of the optimal solution obtained by an algorithm, the true/reference optimal solution should be determined beforehand to calculate the difference between the optimal solution of the algorithm and the true/reference optimal solution in the objective space. In this study, the true optimum was identified in advance using the brute-force search method. The specific steps are described as follows. (1) The one-factor-at-a-time method (i.e., varying only one variable while keeping the others unchanged) was used to explore the impact of each design variable on the annual energy consumption of the reference building. The results indicated that the variables of the wall insulation conductivity and the window upper positions in each façade all had negative impacts on the annual energy consumption. That is, the annual energy consumption of the reference building will increase as the values of these variables increase. Therefore, the best value for these variables should be the lower bounds of their ranges. (2) However, the relationship between the annual energy consumption and the other two optimization variables, namely, building orientation and cooling supply air temperature, were found to be nonmonotonic. Therefore, an exhaustive search was conducted by discretizing their ranges using the step size listed in Table 1. Specifically, the values of the other five variables maintained the lower bounds of their ranges. As a result, the objective function values of 1581 solutions were simulated using EnergyPlus and compared. Finally, the true optimum for the representative optimization problem was identified as X^*=(0.01, 180, 1.35, 1.35, 1.35, 1.35, 12.3). The corresponding objective function value of X^* was simulated as 144.15 kW h/m^2·a. The desired accuracy level δ_* of satisfactory optimal solutions was set at 1% in this study.

Note that the cost of the brute-force search method is proportional to the number of candidate solutions, which in many practical problems tends to grow very quickly as the size of the problem increases. Therefore, brute-force search is typically used when the problem size is limited or when there are problem-specific heuristics that can be used to reduce the set of candidate solutions to a manageable size. All of these limitations of the brute-force search method promote the use of optimization algorithms in BEO. For those cases that the true optimum cannot be determined by the brute-force method, a reference solution that is acceptable by the designer can be used in place of the true optimum to compare the quality of the optimal solution obtained by an algorithm.

5.2. Algorithm control parameters

All the control parameters involved in each algorithm are listed in Table 2. Note that only a handful of these control parameters are randomly chosen for each algorithm in each test, which are underlined in Table 2. For the discrete Armijo gradient algorithm, the alpha and beta parameters, which control the convergence, are of particular importance and are therefore randomly set. For the Hooke-Jeeves algorithm, the initial mesh size exponent, which affects the search step size, is the key control parameter. For the PSO algorithms, the cognitive acceleration and social acceleration parameters control the cognitive behavior and the social behavior of particles, respectively. The values of these two parameters are randomly selected to investigate whether or not they cause the algorithm to be ineffective. Other control parameters for each selected algorithm are set to values that have been demonstrated to perform well for energy-related building design optimization problems [35]. The authors conducted pretests to confirm these settings.

Table 3 provides the results for the trial optimizations (each statistic represents 10 repeated optimization runs), as well as the reliability of each algorithm on each test calculated using Eq. (2). In Fig. 10, each algorithm has 4 consecutive boxplots, corresponding to the 4 tests listed in Table 2. Each boxplot shows the quality variation of the final optimization results generated by running the same optimization test 10 times. Notably, to avoid the influence of different initial solutions on the algorithm's performance evaluation results, all optimization runs use the same initial solution listed in Table 1. Specifically, the quality of the selected initial solution is high, as it is located near the global optimum in the design space.

Fig. 10 shows that when the alpha and beta parameters of the discrete Armijo gradient algorithm were set at different values, the quality of the final optimization results varied correspondingly. Specifically, on Tests 1, 2 and 3, the algorithm successfully converged to a satisfactory optimal solution in each repeated run within the time constraint, with a quality that met the desired accuracy requirements. However, on Test 4, the algorithm searched so slowly that it completed searching after 1718 simulations, which greatly exceeds the given time budget (300 simulations). Consequently, the optimal solution obtained in the first 300 simulations

Table 2
Algorithm control parameter settings for each test.

Common Termination criteria for all algorithms				
Maximum iterations	2000			
Maximum equal results	100			
Maximum simulations	300			
	Test 1	**Test 2**	**Test 3**	**Test 4**
Discrete Armijo gradient				
Alpha	0.7	0.2	0.5	0.9
Beta	0.6	0.3	0.8	0.9
Gamma	0.1	0.1	0.1	0.1
K0	0	0	0	0
KStar	−10	−10	−10	−10
LMax	50	50	50	50
Kappa	25	25	25	25
EpsilonM	0.01	0.01	0.01	0.01
EpsilonX	0.5	0.5	0.5	0.5
Hooke-Jeeves				
Mesh size divider	2	2	2	2
Initial mesh size exponent	0	1	2	3
Mesh size exponent increment	1	1	1	1
Number of step reductions	4	4	4	4
PSO				
Neighborhood topology	Von Neumann	Von Neumann	Von Neumann	Von Neumann
Neighborhood size	5	5	5	5
Number of particles	5	5	5	5
Number of generations	35	35	35	35
Seed	0–9	0–9	0–9	0–9
Cognitive acceleration	2.8	2.3	1.8	1.3
Social acceleration	1.3	1.8	2.3	2.8
Maximum velocity gain continuous	0.5	0.5	0.5	0.5
Max velocity discrete	4	4	4	4
PSOCC				
Constriction gain	0.5	0.5	0.5	0.5
PSOIW				
Initial inertia weight	1.2	1.2	1.2	1.2
Final inertia weight	0	0	0	0

Table 3
Optimal solutions obtained by each algorithm in each test.

Algorithms	Tests	Objective values (kW·h/m^{2}·a)				Reliability β (%)
		Minimum	Maximum	Mean	Standard Deviation	
PSOCC	Test 1	150.74	157.82	154.12	2.45	0
	Test 2	149.16	156.91	153.01	2.70	0
	Test 3	147.88	155.90	151.44	2.68	0
	Test 4	146.94	155.27	150.40	2.47	0
PSOIW	Test 1	144.26	145.83	144.69	0.49	90
	Test 2	144.21	145.86	144.73	0.51	90
	Test 3	144.21	145.48	144.78	0.41	100
	Test 4	144.31	146.28	144.98	0.61	90
Discrete Armijo gradient	Test 1	144.53	144.53	144.53	0	100
	Test 2	144.89	144.89	144.89	0	100
	Test 3	144.73	144.73	144.73	0	100
	Test 4	158.28	158.28	158.28	0	0
Hooke-Jeeves	Test 1	144.18	144.18	144.18	0	100
	Test 2	144.16	144.16	144.16	0	100
	Test 3	144.16	144.16	144.16	0	100
	Test 4	144.32	144.32	144.32	0	100

is located so far from the true optimum that the relative difference between their objective function values is considerably larger than the desired accuracy level. Since the discrete Armijo gradient algorithm belongs to the family of deterministic algorithms, the 10 repeated optimization runs on each test yielded the same results. As a result, the reliability of the algorithm is 100% for Tests 1, 2 and 3 and is 0 for Test 4. Therefore, the discrete Armijo gradient algorithm is considered effective for Tests 1, 2 and 3, and ineffective for Test 4. It can be concluded that the alpha and beta parameters are possible factors that may cause the discrete Armijo gradient algorithm to be ineffective. Setting a large value for both may slow the search speed, and as a result, affect the quality of the optimization result and lead to the failure of the algorithm.

As shown in Fig. 10, for the Hooke-Jeeves algorithm, although its initial search step size varied randomly for the four tests, the algorithm could consistently find satisfactory optimal solutions

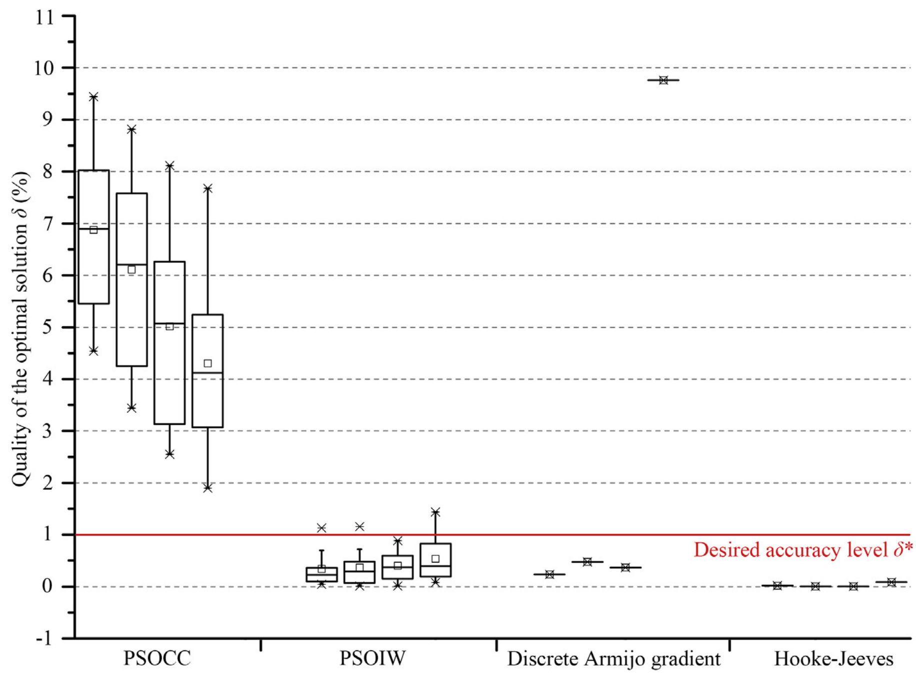

Fig. 10. Quality variability of the optimal solutions obtained by each algorithm in each test with different algorithm parameter settings.

whose quality meets the required level of accuracy. Because the Hooke-Jeeves algorithm, like the discrete Armijo gradient algorithm, is a deterministic algorithm, it achieves a reliability of 100% on each test. Thus, the search step size parameter of the Hooke-Jeeves algorithm has nothing to do with its effectiveness or ineffectiveness when the quality of the initial solution is good.

As shown in Fig. 10, for the PSOCC algorithm, the quality of the best solution found in each optimization run always failed to meet the required accuracy level, regardless of the values of the algorithm's convergence-related parameters (i.e., cognitive acceleration and social acceleration) for the four tests. As a result, the reliability of the PSOCC algorithm is 0 for the four tests, and the algorithm is considered ineffective in these cases. At this point, although it is not certain whether the cognitive acceleration and social acceleration parameters will cause the PSOCC algorithm to be ineffective, it can be concluded that the effectiveness of the PSOCC algorithm cannot be improved by changing these two parameters' values.

Compared with the PSOCC algorithm, the PSOIW algorithm is more effective. On Tests 1, 2 and 4, the algorithm successfully searched for the optimal solutions that satisfy the desired accuracy requirement in nine of ten repeated runs, and consequently, it achieved a reliability of 90% for the three tests, which is acceptable in this study. Moreover, on Test 3, the reliability of the PSOIW algorithm was 100%, because the quality of the optimal solutions obtained in ten repeated runs is good enough to satisfy the desired accuracy level. Therefore, the performance of the PSOIW algorithm is insensitive to the parameters of cognitive acceleration and social acceleration, and the algorithm remains effective for different values of the two parameters.

5.3. *Position of the initial solution in the design space*

The position of the initial solution in the design space is another possible factor that may cause the ineffectiveness of certain algorithms. The representative optimization problem studied in this paper is a multimodal problem. In addition to the global optimum, several local optimal solutions exist in the design space, and these may cause difficulty for the algorithms when attempting to converge to the global optimum. As shown in Table 4, the performance of the selected algorithms is assessed starting from four different initial solutions. Specifically, the initial solutions used on Test 1 and Test 2 are located in valleys with local optimal traps of poor quality. In contrast, the initial solutions in Test 3 and Test 4 are located far away from local optima. In particular, the initial solution used in Test 3 is near the global optimal solution, and the one in Test 4 is located at the boundary of the design space. Note that the PSO algorithms are population-based algorithms. To ensure the consistency of the experiments, the initial solutions specified in Table 4 were artificially contained in the first generations of the PSO algorithms for each test. Moreover, in all tests, the control parameters of each algorithm are set to the values listed in Test 1 of Table 2, which have been verified to perform well for the representative optimization problem as shown in Fig. 10. Specifically, the parameter settings of the PSO algorithms listed in Test 1 of Table 2 are the default values used in GenOpt.

The best solutions obtained by each algorithm in each test, as well as the algorithm's reliability as calculated by Eq. (2), are summarized in Table 5. Fig. 11 illustrates the quality variability of optimal solutions found by each algorithm in each test. As shown, the discrete Armijo gradient algorithm is effective on Test 3 (with a reliability of 100%) but ineffective on Tests 1, 2 and 4 (with a reliability of 0)—even though all of the optimization runs completed

Table 4
Initial solution used for each test.

Design variables	Initial value			
	Test 1	Test 2	Test 3	Test 4
x_1	0.1	0.1	0.07	0.1
x_2	60	90	150	180
x_3	2	1.7	2.7	2.7
x_4	2.7	2.6	1.4	2.7
x_5	2.5	2	1.9	2.7
x_6	2.3	1.35	2.1	2.7
x_7	18	15	12	10

Table 5
Optimal solutions obtained by each algorithm in each test.

Algorithms	Tests	Objective values (kW·h/m^2·a)				Reliability β (%)
		Minimum	Maximum	Mean	Standard Deviation	
PSOCC	Test 1	150.56	157.73	153.94	2.60	0
	Test 2	150.75	157.78	154.06	2.53	0
	Test 3	150.75	157.77	154.09	2.46	0
	Test 4	150.16	157.68	153.83	2.61	0
PSOIW	Test 1	144.22	145.38	144.58	0.42	100
	Test 2	144.21	145.40	144.65	0.45	100
	Test 3	144.27	147.61	145.05	1.01	90
	Test 4	144.21	147.29	144.86	0.90	90
Discrete Armijo gradient	Test 1	146.53	146.53	146.53	0	0
	Test 2	146.09	146.09	146.09	0	0
	Test 3	144.46	144.46	144.46	0	100
	Test 4	149.43	149.43	149.43	0	0
Hooke-Jeeves	Test 1	145.74	145.74	145.74	0	0
	Test 2	145.74	145.74	145.74	0	0
	Test 3	144.15	144.15	144.15	0	100
	Test 4	144.15	144.15	144.15	0	100

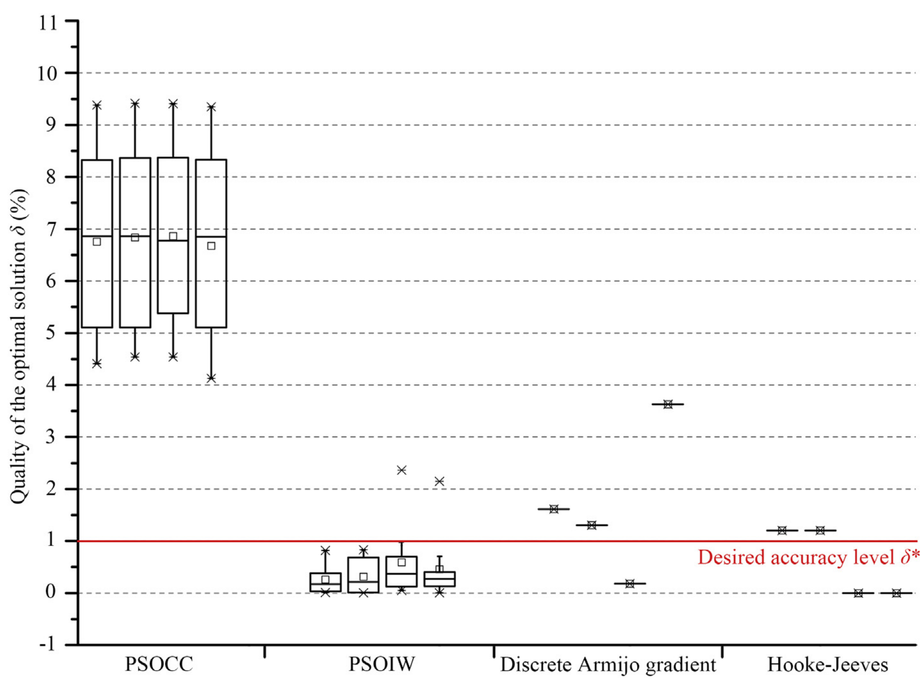

Fig. 11. Quality variability of the optimal solutions obtained by each algorithm in each test with different initial solutions.

convergence successfully within the 300-simulations limit. In Test 1, in which the initial solution was located near a local optimum, although the algorithm managed to escape the local optimum, it still failed to find a solution that satisfied the desired accuracy level. In Test 2, the discrete Armijo gradient algorithm became trapped by the local optimum and failed to converge to the global optimum. In Test 4, the algorithm failed to converge in the late search stage; consequently, the quality of its optimal solution was far from the required accuracy level. In contrast, the algorithm consistently obtained satisfactory solutions for Test 3, in which the initial solution was near the global optimum. Therefore, the initial solution is an important factor in the effectiveness and efficiency of the discrete Armijo gradient algorithm. Using a poor initial solution (e.g., those near local optima or borders) can cause the algorithm to fall into local optima traps and obstruct its convergence process.

Fig. 11 shows that the Hooke-Jeeves algorithm performs well on Tests 3 and 4 when the initial solutions are far from local optima, but poorly on Tests 1 and 2 when the initial solutions are located near local optima. This outcome corresponds with the operational strategy of the Hooke-Jeeves algorithm, which starts from the initial solution and fine-tunes the optimization process to locate a better solution. When the algorithm finds the best solution within a neighboring set of candidate solutions, it can easily mistake that solution as the global optimum and, consequently, fail to explore other portions of the design space. Therefore, the performance of the Hooke-Jeeves algorithm is highly sensitive to the position of the initial solution in the design space. Poor positions of the initial solution (e.g., those located near local optima) may cause the algorithm to be ineffective because it is vulnerable to local optima and, finally, fails to find a satisfactory optimal solution.

As shown in Fig. 11, the PSOCC algorithm fails on the four tests because the quality of the best solutions found by the algorithm in each optimization run is far from the required accuracy level. As a result, the reliability of the algorithm on the four tests is 0. Thus,

Table 6
Suggestions for the selected algorithms to avoid ineffectiveness.

Algorithms	Algorithm control parameter settings	Selection of the initial solution
Discrete Armijo gradient	The alpha and beta parameters need to be set exactly while avoiding that both are set to a large value.	The initial solution is best located near the global optimum.
Hooke-Jeeves	The search step size of the Hooke-Jeeves algorithm is not required to be exactly set as it has nothing to do with the ineffectiveness of the algorithm.	The initial solution is best located far from local optima.
PSOIW	The cognitive acceleration and social acceleration parameters are not required to be exactly set as they have little impact on the effectiveness of the algorithm.	There is no restriction on the selection of the initial solution.
PSOCC	The PSOCC algorithm is not recommended for BEO problems.	

regardless of where the initial solution is located in the design space, the PSOCC algorithm is ineffective for the representative optimization problem.

In contrast to the PSOCC algorithm, the PSOIW algorithm consistently obtained satisfactory optimal solutions that met the accuracy requirement within the given time limit with a reliability of 100% for Tests 1 and 2 and 90% for Tests 3 and 4. Thus, the performance of the PSOIW algorithm is independent of the initial solution. Regardless of the location of the initial solution—even if it is located near a local optimum—it will not cause the algorithm to fail.

5.4. Suggestions for the selected algorithms to avoid ineffectiveness

After studying the ineffectiveness and possible causes of the four selected optimization algorithms, some natural and meaningful suggestions can be given for these algorithms to avoid ineffectiveness when they are used for BEO problems. As shown in Table 6, these suggestions are provided from two perspectives: algorithm control parameter settings and selection of the initial solution.

6. Conclusions

Optimization algorithms play a critical role in determining the effectiveness and efficiency of BEO techniques. An effective optimization algorithm should simultaneously meet three criteria: (1) deliver a satisfactory optimal solution, (2) complete the optimization process within a given time constraint, and (3) achieve good reliability. When an optimization algorithm violates any of these three criteria, it is considered ineffective.

In BEO, an optimization algorithm is vulnerable to being ineffective when it exhibits one or more of the following symptoms: (1) becoming trapped in local optima, (2) having insufficient search speed, (3) failing to converge, (4) terminating before convergence because of various errors, or (5) having poor reliability.

To investigate the causes of ineffectiveness, four commonly used optimization algorithms were selected and applied to a representative BEO problem to perform numerical experiments. The results demonstrate that the following causes led to the ineffectiveness of these algorithms.

- The effectiveness of the discrete Armijo gradient algorithm is dependent on its control parameters and the position of the initial solution in the design space. Inappropriate settings of the alpha and beta parameters can slow the search speed and eventually cause the algorithm to fail to converge within the predetermined time constraint. In addition, when the initial solution is located far from the global optimum, the algorithm requires more time to reach the global optimum, which can result in ineffectiveness when the required time exceeds the allotted time constraint. Specifically, when the initial solution is located near local optima, it is difficult for the algorithm to escape.
- For the Hooke-Jeeves algorithm, the position of the initial solution in the design space is crucial to the effectiveness of the algorithm because of its poor ability to escape from local optima. Moreover, changing its search step size has no impact on the effectiveness and efficiency of the algorithm.
- The PSOIW algorithm showed the best performance in all tests. The two control parameters (i.e., cognitive acceleration and social acceleration) and the position of the initial solution in the design space had little impact on the effectiveness of the PSOIW algorithm.
- Unlike the PSOIW algorithm, the PSOCC algorithm was ineffective for all tests: it never found a satisfactory optimal solution, regardless of the algorithm's convergence-related parameter settings (i.e., cognitive acceleration and social acceleration) or the initial solution. Because the essential difference between the PSOCC algorithm and the PSOIW algorithm is the coefficient methods used to constrain the search of a PSO algorithm, the constriction coefficient method is not recommended for a PSO algorithm applied to solve BEO problems.

7. Future study

One of the primary objectives of this study was to evaluate whether different algorithm control parameter settings and different initial solutions can cause the four studied algorithms to be ineffective when used in BEO. Therefore, four sets of different control parameter settings and four different initial solutions were randomly chosen for each test. However, the best parameter settings and the best initial solution for each algorithm are beyond the scope of this paper and are therefore not provided. Such work is valuable and will be pursued in future research.

Acknowledgments

This paper was financially supported by the National Natural Science Foundation of China (grant number 51678124) and the Scientific Research Foundation of Graduate School of Southeast University (grant number YBJJ1702).

References

[1] U.S. Energy Information Administration, International Energy Outlook, 2016.
[2] R. Evins, A review of computational optimisation methods applied to sustainable building design, Renew. Sustain. Energy Rev. 22 (8) (2013) 230–245.
[3] A.T. Nguyen, S. Reiter, P. Rigo, A review on simulation-based optimization methods applied to building performance analysis, Appl. Energy 113 (6) (2014) 1043–1058.
[4] M. Wetter, J. Wright, Comparison of a generalized pattern search and a genetic algorithm optimization method, in: Proceedings of the 8th International Building Performance Simulation Association Conference, 2003, pp. 1401–1408.
[5] B. Si, Z. Tian, X. Jin, X. Zhou, P. Tang, X. Shi, Performance indices and evaluation

of algorithms in building energy efficient design optimization, Energy 114 (2016) 100–112.

[6] A. Lorestani, M.M. Ardehali, Optimization of autonomous combined heat and power system including PVT, WT, storages, and electric heat utilizing novel evolutionary particle swarm optimization algorithm, Renew. Energy 119 (2018) 490–503.

[7] R. Pereira, L. Aelenei, Optimization assessment of the energy performance of a BIPV/T-PCM system using genetic algorithms, Renew. Energy (2018).

[8] A. Baniassadi, M. Shirinbakhsh, F. Torabi, Multivariate optimization of off-grid wind turbines with variable demand-Case study of a remote commercial building, Renew. Energy 101 (2017) 1021–1029.

[9] X. Shi, Z. Tian, W. Chen, B. Si, X. Jin, A review on building energy efficient design optimization from the perspective of architects, Renew. Sustain. Energy Rev. 65 (2016) 872–884.

[10] Y.C. Ho, D.L. Pepyne, Simple explanation of the No-Free-Lunch theorem and its implications, J. Optim. Theor. Appl. 115 (3) (2002) 549–570.

[11] J.J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.N. Suganthan, A. Carlos, et al., Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization, Int. J. Comput. Assist. Radiol. Surg. (2) (2006).

[12] J.J. Liang, B.Y. Qu, P.N. Suganthan, A.G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-parameter Optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013, pp. 3–18. Technical Report 201212.

[13] C.J. Hopfe, Uncertainty and Sensitivity Analysis in Building Performance Simulation for Decision Support and Design Optimization, PhD Diss, Eindhoven University, Eindhoven, 2009.

[14] M. Hamdy, A.T. Nguyen, J.L.M. Hensen, A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems, Energy Build. 121 (6) (2016) 57–71.

[15] K. Sundareswaran, S. Palani, Application of a combined particle swarm optimization and perturb and observe method for MPPT in PV systems under partial shading conditions, Renew. Energy 75 (2015) 308–317.

[16] M. Sharafi, T.Y. ElMekkawy, E.L. Bibeau, Optimal design of hybrid renewable energy systems in buildings with low to high renewable energy ratio, Renew. Energy 83 (2015) 1026–1042.

[17] J.J. Wang, Z.L. Xu, H.G. Jin, G.H. Shi, C. Fu, K. Yang, Design optimization and analysis of a biomass gasification based BCHP system: a case study in Harbin, China, Renew. Energy 71 (2014) 572–583.

[18] L. Magnier, F. Haghighat, Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and artificial neural network, Build. Environ. 45 (3) (2010) 739–746.

[19] M.A.I. Khan, C.J. Noakes, V.V. Toropov, Multi-objective optimization of the ventilation system design in a two-bed ward with an emphasis on infection control, in: Proceedings of the 2012 Building Simulation and Optimization Conference, Loughborough, Leicestershire, 2012, pp. 9–18.

[20] G.H. Mcclelland, W.D. Schulze, D.L. Coursey, Insurance for Low-probability Hazards: a Bimodal Response to Unlikely Events. Making Decisions about Liability and Insurance, Springer Netherlands, 1993.

[21] W. Forst, D. Hoffmann, Optimization-theory and Practice, Springer, New York, 2010.

[22] M. Wetter, E. Polak, A convergent optimization method using pattern search algorithms with adaptive precision simulation, Build. Serv. Eng. 25 (4) (2004) 327–338.

[23] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, et al. US Department of Energy commercial reference building models of the national building stock. http://www.nrel.gov/docs/fy11osti/46861.pdf (accessed 10 January 2018).

[24] 2003 Commercial Buildings Energy Consumption Survey. Washington, DC: EIA. https://www.eia.gov/consumption/commercial/data/2003/(accessed 10 January 2018).

[25] D.W. Winiarski, M.A. Halverson, W. Jiang, DOE's commercial building benchmarks-development of typical construction practices for building envelope and mechanical systems from the 2003 CBECS, in: Proceedings of 2008 Summer Study on Energy Efficiency in Buildings, American Council for an Energy Efficient Economy.

[26] American Society of Heating, Refrigerating and Air-conditioning Engineers. Energy Standard for Buildings except Low-rise Residential Buildings. ANSI/ASHRAE/IESNA Standard 90.1-2004. Atlanta, Georgia..

[27] D.B. Crawley, L.K. Lawrie, F.C. Winkelmann, W.F. Buhl, Y.J. Huang, C.O. Pedersen, R.K. Strand, R.J. Liesen, D.E. Fisher, M.J. Witte, J. Glazer, EnergyPlus: creating a new-generation building energy simulation program, Energy Build. 33 (4) (2001) 319–331.

[28] J. Amaya, The armijo step rule adapted to gradient path algorithms, in: B.L. Contesse, F.R. Correa, P.A. Weintraub (Eds.), Recent Advances in System Modelling and Optimization, Lecture Notes in Control and Information Sciences, vol. 87, Springer, Berlin, Heidelberg, 1986, pp. 1–6.

[29] R. Hooke, T.A. Jeeves, "Direct search" solution of numerical and statistical problems, J. ACM 8 (2) (1961) 212–229.

[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.

[31] M. Wetter, GenOpt® Generic Optimization Program User Manual Version3.1.0, Lawrence Berkeley National Laboratory, 2011. http://simulationresearch.lbl.gov/GO/download/manual-3-1-0.pdf. (Accessed 10 January 2018).

[32] G.B. Park, M. Jeong, D.H. Choi, A guideline for parameter setting of an evolutionary algorithm using optimal Latin hypercube design and statistical analysis, Int. J. Precis. Eng. Manuf. 16 (10) (2015) 2167–2178.

[33] https://simulationresearch.lbl.gov/GO/download.html (accessed 10 January 2018)..

[34] https://energyplus.net/downloads (accessed 10 January 2018).

[35] M. Wetter, J. Wright, A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization, Build. Environ. 39 (8) (2004) 989–999.