

7th International Building Physics Conference

IBPC2018

Proceedings

SYRACUSE, NY, USA

September 23 - 26, 2018

Healthy, Intelligent and Resilient
Buildings and Urban Environments

ibpc2018.org | [#ibpc2018](https://twitter.com/ibpc2018)



Criteria for identifying failure optimization algorithms in building energy optimization and case studies

Binghui Si¹ and Xing Shi^{1,*}

¹School of Architecture, Southeast University, Nanjing, China

*Corresponding email: shixing_seu@163.com

ABSTRACT

Optimization algorithms play a vital role in the Building Energy Optimization (BEO) technique. Although many algorithms are currently used in BEO, it is difficult to find an algorithm that performs well for all optimization problems. Some algorithms may fail in some cases. This study specifically focuses on failure algorithms in BEO and the possible causes. Several criteria are proposed for identifying failure algorithms. Four optimization problems based on the DOE small and large office buildings are developed. Three commonly used algorithms in BEO, namely, Pattern Search (PS) algorithm, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm, are applied to the four problems to investigate possible reasons for their failure. Results indicate that the effectiveness of the three selected algorithms is highly dependent on the optimization problems to be addressed. Besides, the control parameter setting of the PS algorithm appears to be a significant factor that may cause the algorithm to lose effectiveness. However, it does not seem to be the main reason for the failure of the GA and PSO algorithm. In General, the results gained from this study can deepen our understanding of optimization algorithms used in BEO. Besides, understanding the reasons why optimization algorithms are ineffective can help architects, engineers, and consultants select the appropriate optimization algorithms and set their parameters to achieve a better BEO design that is less vulnerable to failure.

KEYWORDS

Building energy optimization; failure optimization algorithm; cause of failure; algorithm parameter setting

INTRODUCTION

Building Energy Optimization (BEO) is a booming technique that combines building energy simulation engines with optimization engines. Unlike the conventional “trial-and-error” design methodology, which requires designers to manually adjust the design based on their experience and limited simulations, the BEO technique can automatically generate and simulate new designs utilizing optimization algorithms and performance simulation software and finally achieve the best design based on the predefined design objectives (Si et al. 2016). Therefore, optimization algorithms play a crucial role in the application of the BEO technique.

As shown in some important review works (Machairas et al. 2014; Shi et al. 2016), a quite number of algorithms can be used in BEO, for example, the evolutionary algorithms, direct search algorithms, hybrid algorithms, etc. However, there is in fact no universal algorithm that applies to all optimization problems, which means an algorithm may fail under certain circumstances. Thus, finding the causes for their failure and exploring the circumstances under which an algorithm may become fail can significantly help designers to choose an appropriate algorithm among the available options and help them avoid failure algorithms.

The objective of this research is to study failure optimization algorithms used in BEO and possible failure reasons. The first research task is to develop a set of criteria to recognize whether an optimization algorithm fails for a BEO problem. Then four optimization problems are developed using the DOE small and large office buildings. Three optimization algorithms are selected to investigate the possible factors that may cause them to fail for the four optimization problems.

METHODS

Criteria for identifying failure algorithms

Before defining a failure algorithm, we need to distinguish two concepts: a failure algorithm and a failure optimization run. For a specific optimization problem, an optimization algorithm fails on an optimization run does not mean it fails for the optimization problem. The reasons are stochastic optimization algorithms (e.g., GA, PSO, etc.) usually involve random operators in their optimization processes, which will result in different optimization runs when they are run repeatedly. In this case, one specific optimization run cannot reflect the performance of the algorithm. Users need to repeat the optimization test as many times as possible and then analyse all optimization runs. However, for a determined optimization algorithm, it usually has a unique optimization run which can fully reflect the performance behaviour of the algorithm when all relevant parameters and the initial solution remain unchanged. Therefore, in this study, we firstly proposed two criteria, which are the most concerned issues for designers when using optimization techniques, to identify a failure optimization run. Then the failure rate criterion was used to identify a failure algorithm. Note that this paper is particularly focuses on single-objective algorithms because about 60% of the building optimization studies used the single-objective approach (Nguyen et al. 2014). Multi-objective optimization algorithms are not covered.

In general, a successful optimization run should find the optimal solution within the desired accuracy level using a limited amount of time. It requires two criteria that should be met simultaneously, one of which is the quality of the optimal solution obtained in the optimization run should be high enough to meet the users' requirements, and the other is the computing time cannot exceed the time limit. An optimization run that violates any of the above two criteria is considered failure. In this study, to measure the quality of the optimal solution, Equation 1 can be used to calculate the relative distance between the optimal solution found in an optimization run and the true optimum of the optimization problem.

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

where $f(X')$ is the objective value of the optimal solution found in an optimization run, and $f(X^*)$ is the objective value of the true optimum, which in some cases can be obtained through brute-force search. If the value of δ is larger than that of δ^* which is the acceptable accuracy level defined by the designer, then the optimization run is considered failure.

To define a failure optimization algorithm for a given problem, the algorithm needs to repeat the optimization process several times and then those failure runs need to be isolated to calculate the failure rate, which in essence, is the ratio of failure optimization runs to the total runs. Equation 2 provides a formula.

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

where $N_{failure}$ is the number of failure runs and N_{total} is the total number of runs driven by the algorithm. According to the Low Probability Event (LPE) principle (McClelland et al. 1993), which is an important theorem in probability and commonly applied in practical projects and mathematical statistics, an LPE is considered will not occur in the actual environment. In practice, the value of 0.01, 0.05 or 0.1 are commonly used for an LPE which can be denoted by β^* . Users can also set other values according to their specific conditions. Consequently, in this study, an algorithm is considered failure for a given optimization problem when $\beta > \beta^*$.

Description of the standard optimization problem

Table 1. Specifications of optimization variables.

Design variables	Symbol	Unit	Step size	Range	Initial value
Building long axis azimuth	x_1	°	5	[0,180]	90
Cooling set-point temperature	x_2	°C	0.05	[22,29]	24
Heating set-point temperature	x_3	°C	0.05	[15,22]	21
Roof insulation conductivity	x_4	W/m·K	0.001	[0.03,0.06]	0.049
Roof insulation thickness	x_5	m	0.002	[0.01,0.15]	0.126
South wall insulation conductivity	x_6	W/m·K	0.001	[0.03,0.06]	0.049
East wall insulation conductivity	x_7	W/m·K	0.001	[0.03,0.06]	0.049
North wall insulation conductivity	x_8	W/m·K	0.001	[0.03,0.06]	0.049
West wall insulation conductivity	x_9	W/m·K	0.001	[0.03,0.06]	0.049
South wall insulation thickness	x_{10}	m	0.002	[0.01,0.15]	0.036
East wall insulation thickness	x_{11}	m	0.002	[0.01,0.15]	0.036
North wall insulation thickness	x_{12}	m	0.002	[0.01,0.15]	0.036
West wall insulation thickness	x_{13}	m	0.002	[0.01,0.15]	0.036
South window upper position	x_{14}	m	0.02	[1,2.7]	2.5
East window upper position	x_{15}	m	0.02	[1,2.7]	2.5
North window upper position	x_{16}	m	0.02	[1,2.7]	2.5
West window upper position	x_{17}	m	0.02	[1,2.7]	2.5
South window U-value	x_{18}	W/m ² K	0.05	[1,7]	3.25
East window U-value	x_{19}	W/m ² K	0.05	[1,7]	3.25
North window U-value	x_{20}	W/m ² K	0.05	[1,7]	3.25

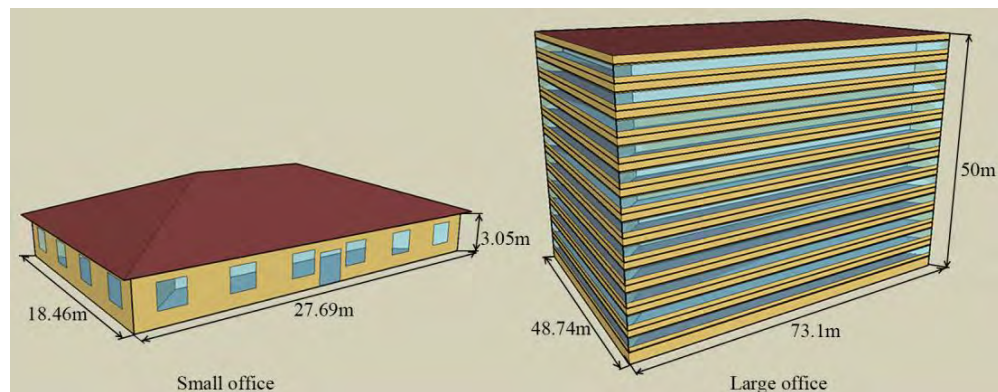


Figure. 1 Perspective views of the DOE small and large office buildings.

In this study, four optimization problems with 10 and 20 optimization variables respectively were developed following the models of the DOE small and large office buildings (Deru et al. 2011) in Baltimore, USA. They were all designed to minimize the annual energy consumption of the case buildings. Figure 1 shows the architectural schematic views of the buildings. They

all have one core thermal zone and four perimeter thermal zones on each floor. Table 1 lists the optimization variables involved in the optimization problems as well as their initial values, step sizes and range of variations. Specifically, the value for the lower window position is fixed at 0.9 m, and the windows in the same facade are of equal area. Besides, the first ten variables were used for optimization problems with 10 design variables.

RESULTS

In this section, three commonly used optimization algorithms in BEO, namely, Pattern Search (PS) algorithm, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm were assessed to find out possible factors that may cause the algorithms to fail. As shown in Table 2, four algorithm parameter settings for each algorithm are randomly generated to investigate their impacts on the effectiveness of the selected algorithms. Readers are referred to the manual book (Wetter M, 2011) for more information of the working strategies and the original development of each algorithm. Based on the two criteria proposed above about a failure optimization run, two evaluation approaches are accessible: (1) assessing the quality of the optimal solution obtained when the computing time is restricted; (2) assessing the computing time consumed when the optimization run finds the desired solution. In this study, we chosen the first approach. The maximum number of simulations for each optimization run was restricted at 300, and each optimization process was repeated 10 times to calculate the failure rate. These numbers were chosen to strike a balance between what is preferred and what is practical in terms of computing time. Specifically, the true optimum of the four optimization problems were obtained by brute-force search and were listed in Table 3. Besides, the desired accuracy level δ^* of optimal solutions was set at 1%, and the acceptable maximum failure rate β^* was 10%.

Table 2. Algorithm control parameter settings for each algorithm.

Algorithms	Parameters	Test 1	Test 2	Test 3	Test 4
PS	Expansion factor	2	3	4	5
	Contraction factor	0.2	0.4	0.6	0.8
GA	Population size	10	15	20	30
	Number of generations	30	20	15	10
	Elite count	1	2	3	4
	Crossover fraction	0.2	0.4	0.6	0.8
	Mutation rate	0.05	0.1	0.15	0.2
PSO	Population size	10	15	20	30
	Maximum number of iterations	30	20	15	10
	acceleration const 1 (local best influence)	2	3	2	3
	acceleration const 2 (global best influence)	2	2	3	3
	Initial inertia weight	0.9	0.8	0.7	0.6
	Final inertia weight	0.4	0.3	0.2	0.1

For each optimization problem, the quality variation of the optimal solution obtained by each algorithm in each test were illustrated in Figure 2. As shown, each algorithm has 4 consecutive boxplots, corresponding to the 4 tests listed in Table 2. It is noted that all optimization runs used the same initial solution listed in Table 1 to avoid the influence of different initial solutions on the evaluation results.

As shown in Figure 2, for each optimization problem, the average quality of optimal solutions found by the PS algorithm changes violently between different tests, which means the performance of the algorithm is sensitive to its parameter settings. It is further verified when the PS algorithm was use to solve Problem 1, in which it succeed in Test 1 but failed in Tests

2-4. Therefore, inappropriate parameter settings of the PS algorithm may cause it to fail. However, for the same optimization problem, the average quality of optimal solutions searched by the GA and PSO algorithm appears to be more stable between different tests. Thus, the effectiveness of the two algorithms are less sensitive to their parameter settings. Although GA and PSO algorithm failed in all four tests for Problem 1, 2 and 4, we cannot conclude if different parameter settings will cause the two algorithm to lose effective.

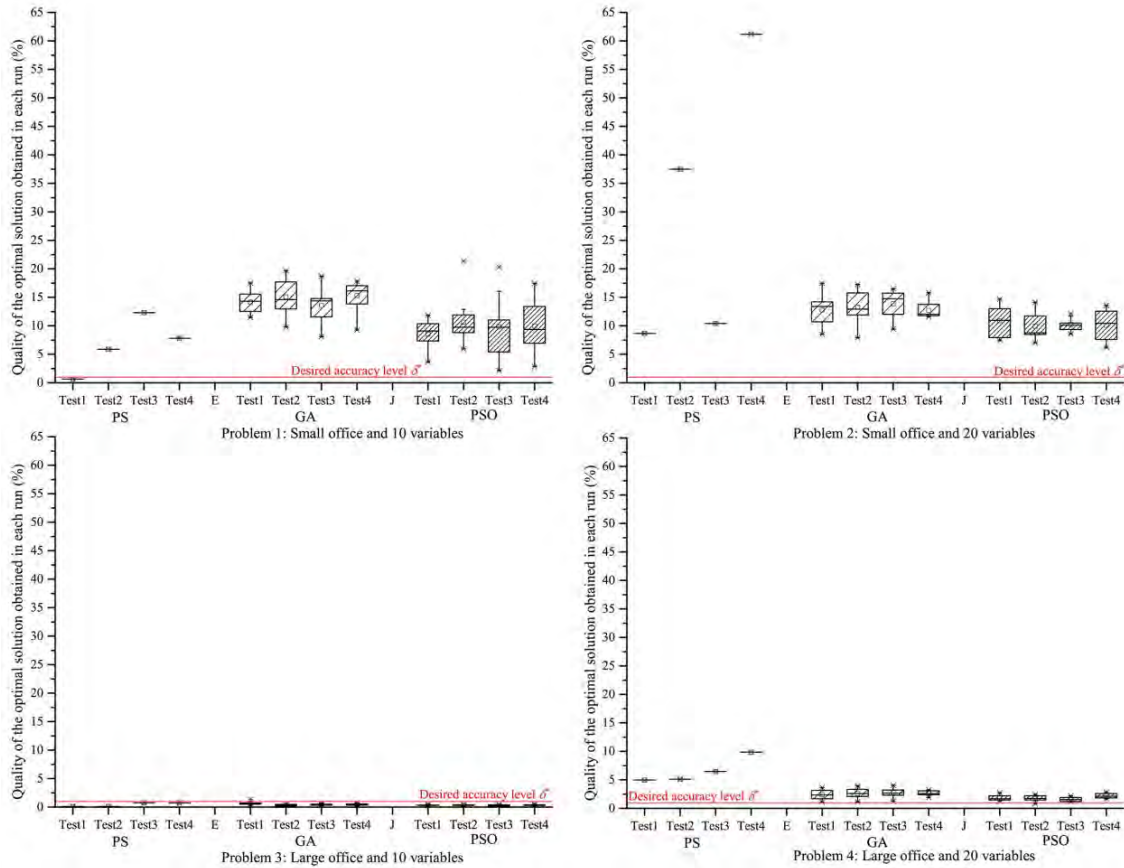


Figure. 2 Quality variability of the optimal solutions obtained by each algorithm in each test with different algorithm parameter settings.

Table 3. True optimum of each optimization problem and failure rate of each algorithm for each test.

Index	Optimization problems	True optimum (kW·h/m ² a)	Algorithms	Failure rate			
				Test 1	Test 2	Test 3	Test 4
Problem 1	Small office and 10 variables	95.532	PS	0	100%	100%	100%
			GA	100%	100%	100%	100%
			PSO	100%	100%	100%	100%
Problem 2	Small office and 20 variables	134.741	PS	100%	100%	100%	100%
			GA	100%	100%	100%	100%
			PSO	100%	100%	100%	100%
Problem 3	Large office and 10 variables	96.066	PS	0	0	0	0
			GA	10%	0	0	0
			PSO	0	0	0	0
Problem 4	Large office and 20 variables	125.355	PS	100%	100%	100%	100%
			GA	100%	100%	100%	100%
			PSO	100%	90%	100%	100%

Table 3 gives the calculated failure rate of the trial optimizations (each statistic relating 10 repeated optimization runs). It shows that for Problem 1, the PS algorithm performed well for Test 1 with a failure rate of 0, but failed for Tests 2-4 with a failure rate of 100%. The quality of the optimal solutions obtained by GA and PSO in the four tests were all beyond the desired accuracy level of Problem 1, and therefore, their failure rates were all 100%. For Problems 3 and 4, all the three algorithms failed to find desired solutions in all tests with a failure rate larger than the acceptable maximum failure rate (i.e., 10%). However, when applying the three algorithms to Problem 2, all of them could consistently find desired optimal solutions with a failure rate of no more than 10% even when they used different parameter settings. Thus, the effectiveness of the three selected algorithms highly depends on the optimization problems solved. In this study, some properties involved in the Problems 1, 2 and 4 seem to dominate the failure of the three selected algorithms.

CONCLUSIONS

Optimization algorithms play a critical role in determining the effectiveness and efficiency of BEO techniques. In this study, the criteria for helping users to detect failure optimization algorithms used for BEO problems are proposed. Four optimization problems were developed to find out possible factors that may cause three commonly used algorithms to fail. The numerical results demonstrate the following failure mechanisms of the selected algorithms: (1) algorithm control parameter setting is an important factor that may cause the PS algorithm to fail but it does not seem to be a key factor that may cause the failure of the GA and PSO algorithm. (2) Some inherent properties of optimization problems may cause the three algorithms to fail because their performance appeared to be highly dependent on the optimization problems addressed. Future research is required to examine the impacts of different properties involved in a BEO problem on the performance behaviour of different optimization algorithms.

ACKNOWLEDGEMENT

This paper was financially supported by the Scientific Research Foundation of Graduate School of Southeast University (grant number: YBJJ1702).

REFERENCES

- Si B., Tian Z., Jin X., Zhou X., Tang P. and Shi X. 2016. Performance indices and evaluation of algorithms in building energy efficient design optimization. *Energy*, 114, 100-112.
- Machairas V., Tsangrassoulis A. and Axarli K. 2014. Algorithms for optimization of building design: A review. *Renewable & Sustainable Energy Reviews*, 31(2), 101-112.
- Shi X., Tian Z., Chen W., Si B. and Jin X. 2016. A review on building energy efficient design optimization from the perspective of architects. *Renewable and Sustainable Energy Reviews*, 65, 872-84.
- Nguyen A.T., Reiter S. and Rigo P. 2014. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 113, 1043-1058.
- Mcclelland G.H., Schulze W.D., and Coursey D.L. 1993. Insurance for Low-Probability Hazards: A Bimodal Response to Unlikely Events. *Making Decisions About Liability And Insurance*, Springer Netherlands.
- Deru M., Field K., Studer D., Benne K., Griffith B., Torcellini P., Liu B., Halverson M., Winiarski D., Rosenberg M. and Yazdanian M. 2011. US Department of Energy commercial reference building models of the national building stock.
- Wetter M. 2011. GenOpt® Generic Optimization Program User Manual Version 3.1.0. Lawrence Berkeley National Laboratory.