PERFORMANCE EVALUATION CRITERIA OF ALGORITHMS IN BUILDING ENERGY EFFICIENT DESIGN OPTIMIZATION

Binghui Si[1,2,*], Zhichao Tian[1,2], Xing Jin[1,2], Xing Shi[1,2], Wenqiang Chen[1,2]

[1]*School of Architecture, Southeast University*
[2]*Key Laboratory of Urban and Architectural Heritage Conservation, Ministry of Education*

ABSTRACT
The mandatory requirement of energy efficiency for green and sustainable buildings has led to the emerging development of the building energy efficient design optimization technique, among which optimization algorithms is vital for the overall effectiveness and efficiency of the technique. This study evaluates algorithms used for building energy efficient design optimization. Four performance criteria, namely, stability, validity, speed, and coverage, are proposed. A standard building and a design optimization problem are developed. Hooke–Jeeves algorithm, Multi-Objective Genetic Algorithm II (MOGA-II), and Multi-Objective Particle Swarm Optimization (MOPSO) algorithm are evaluated by using the proposed performance criteria and standard design problem. Results indicate that there is no algorithm performs best in all four criteria. The Hooke–Jeeves algorithm is trapped in the local optimum with poor validity and coverage. The MOPSO algorithm generally performs better than the other two algorithms except for the speed. The MOGA-II performs slightly worse than the MOPSO algorithm in term of validity and coverage but is still acceptable.

KEYWORDS
Performance evaluation, Optimization algorithm, Energy efficient design, Building optimization

INTRODUCTION
Building energy efficient design optimization is a burgeoning technique that relies on optimization algorithms to generate new designs based on user-defined design objective objectives and energy simulation results. Figure 1 shows the general procedure of the energy efficient design optimization technique for buildings (Shi et al. 2016). It is clear that the optimization and energy simulation engines drive the overall design and optimization process. Optimization algorithms play a key role and generate new designs based on simulation results and predefined design objectives. Thus, the performance of the optimization algorithms is vital for the effectiveness and efficiency of the building

* Corresponding author email: sbh.nju@163.com.

optimization workflow.

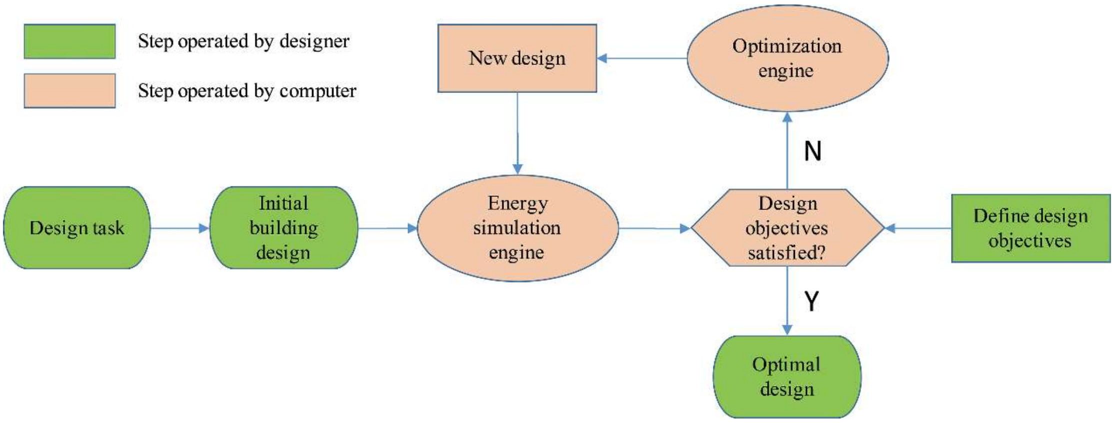

Figure 1. *General procedure of the building energy optimization technique*

The commonly used algorithms in building energy efficient design optimization can be grouped into three categories, namely evolutionary algorithms, derivative-free search algorithms, and hybrid algorithms (Terzidis 2006). As shown in several important review works published lately (e.g. Nguyen et al. 2014), evolutionary algorithms are the most frequently used methods in building performance optimization, accounting for approximately 60%. In particular, the genetic algorithm (GA) and its variations such as non-dominant sorting genetic algorithm (NSGA) are dominant.

Existing literature on the performance of optimization algorithms is mostly from non-architectural fields, including mathematics, computer science, and operations research. For example, different groups of test functions and evaluation criteria are proposed within the context of the IEEE Congress Evolutionary Computation (IEEE-CEC) competitions (Liang et al. 2013), which are of great importance for evaluating and comparing the performance of modified or newly proposed algorithms. Nevertheless, whether these findings hold for energy efficient design optimization problems is still a question. Only few studies shed light on the effectiveness and efficiency of algorithms used in building energy efficient design optimization, e.g. Tuhus-Dubrow and Krarti (2009), and Futrell et al. (2015).

The purpose of this study is to develop a set of criteria to describe the most important performance aspects of algorithms used in building energy efficient design optimization. Following this, a case study is conducted utilizing the criteria proposed.

PERFORMANCE CRITERIA

Stability

The stability is to describe the ability of an algorithm to find the same optimal solution when repeating a given optimization test multiple times. Figure 2 illustrates the comparison of the stability of two algorithms in solving a specific problem. Each point represents the optimal solution obtained in once repeated test. Obviously, the circular solutions are significant different from each other while the triangle solutions are very similar. Thus, Algorithm 1 cannot guarantee consistently finding the same optimal solution and performs worse than Algorithm 2 in terms of stability.

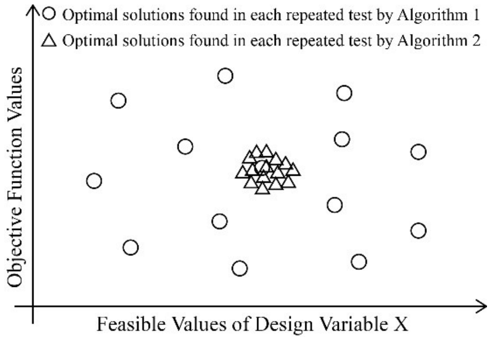

Figure 2. *Comparison of the stability of two algorithms for a specific problem*

To quantitatively compare the stability of algorithms, the standard deviation of the objective function values of optimal solutions obtained through repeating the same test multiple time can be calculated.

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(M-Y_i)^2}{n}}, \quad (1)$$

where

$$M = \frac{\sum_{i=1}^{n} Y_i}{n}, \quad (2)$$

n is the number of repeated tests, and Y_i is the objective function value of the optimal solution obtained in the ith repeated test. For two algorithms, the one with a smaller standard deviation is generally considered more stable.

Validity

The validity is to describe the capability of an algorithm to find the true optimal solution for a given optimization problem within a limited number of objective function evaluations.

The validity can be quantified by examining the closeness of the optimal solution found by an algorithm with the true optimum. Equations of the normalized Euclidean distance $d(X^*, X')$, and the relative distance in their objective function values $g(f(X^*), f(X'))$, are given below.

$$d(X^*, X') = \sqrt{\sum_{k=1}^{m}(\frac{x_k{}' - x_k{}^*}{u_k - l_k})^2}, \quad (3)$$

$$g\big(f(X^*), f(X')\big) = \left|\frac{f(X') - f(X^*)}{f(X^*)}\right| \times 100\%, \quad (4)$$

where $X' = (x_1{}', \ldots\ldots, x_m{}')$ is the optimal solution found by an algorithm;

$X^*=(x_1^*,\ldots\ldots,x_m^*)$ is the true optimal solution; m is the number of problem variables; l_k and u_k are the lower and upper bounds of the kth variable, such that $l_k \le x_k' \le u_k$, $\forall k \in (1,\ldots\ldots,m)$; $f(X)$ is the objective function. Hence, for a given algorithm, if the values of both $d(X^*, X')$ and $g(f(X^*), f(X'))$ are small, the optimal solution found by the algorithm is close to the true optimum, indicating that its validity is high.

Speed

The speed of an algorithm describes its convergence rate to the optimal solution for a given optimization problem. Generally, the computing time of an optimization run is roughly equal to the product of the simulation time of one solution and the total number of solutions explored in the whole run. Thus using a high-speed optimization algorithm which can rapidly converge to the optimal solution can greatly save computing time especially for complex optimization problems, in which the evaluation of the cost function requires a significant amount of time and resources.

The speed of an algorithm can be measured by the number of simulations before the optimal solution is found. For two algorithms, the one converging to its optimal solution with fewer simulations performs better in terms of speed.

Coverage

The coverage of an algorithm describes its ability to search for the optimal solution in the entire feasible solution space. It reflects the global search capability of an algorithm. Coverage can be represented by the distribution of all searched solutions in the feasible region. The wide distribution indicates strong global search ability and good performance in coverage. Figure 3 is the comparison of the distributions of all solutions searched by two algorithms for the same problem. It is obvious that the circular solutions distribute in the feasible region more evenly while the cross solutions concentrate in a certain region. So the diversity of the former is quite better and Algorithm 1 is more likely to find better solutions.

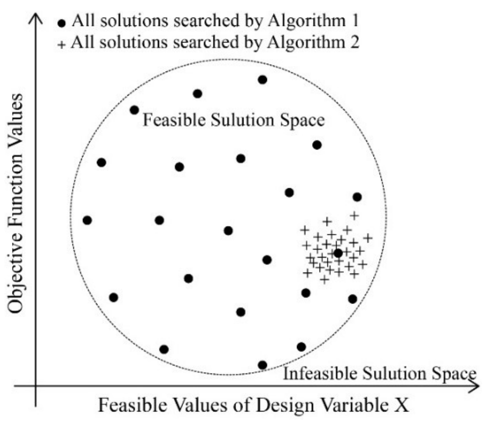

Figure 3. *Illustration of the coverage of two algorithms for a single-variable function*

To quantify the distribution of all searched solutions, the following equations can be used to basically calculate the standard deviation of all searched solutions in each variable domain and their product.

$$COV = \prod_{k=1}^{m} SD_k, \quad (5)$$

where

$$SD_k = \sqrt{\frac{\sum_{j=1}^{t}(M_k - h_{kj})^2}{t}}, \quad (6)$$

and

$$M_k = \frac{\sum_{j=1}^{t} h_{kj}}{t}, \quad (7)$$

COV is the coverage of an algorithm; SD_k is the standard deviation of the *k*th variable values of all searched solutions, $\forall k \in (1,\ldots\ldots,m)$; *m* is the number of independent variables; h_{kj} is the value of the *k*th variable corresponding to the *j*th solution, $\forall j \in (1,\ldots\ldots,t)$; *t* is the number of all solutions searched by the algorithm; and M_k is the mean of the *k*th variable values of all solutions.

NUMERICAL EXPERIMENTS

Standard building

A 3-story rectangular-shaped office building following the model of the DOE medium office building (Anon. A) is used as the standard building in this study. Figure 4 shows the architectural schematic view of the building with an aspect ratio of 1.5. Its total floor area is 4,982 m^2, and the floor height is 4 m. It has one core thermal zone and four perimeter thermal zones on each floor. The HVAC system is a variable air volume flow system with gas furnace and electric reheat. The internal heat gains are composed of lights (10.76 W/m^2), electrical plugs, process loads (10.76 W/m^2), and heat release by the presence of 268 people. No shading devices or overhangs are present in the building.

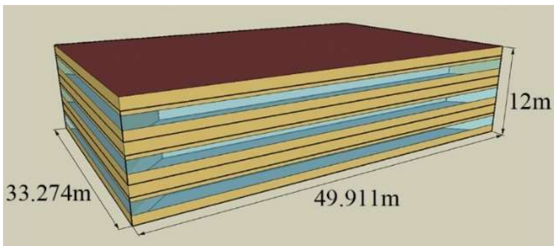

Figure 4. *Projection of the standard building*

Optimization variables and objective

In this study, minimizing building annual energy consumption is the optimization objective to investigate the energy performance of the standard building. The optimization problem consists of six continuous variables, including building orientation, window upper positions in each facade, and wall conductivity. Specifically, the value for the lower window position is fixed at 0.8m, equal to a standard desk height, and the windows in the same facade are of equal area. Table 1 shows the list of optimization variables used in this study as well as their initial value, step size and range of variation.

Table 1. Six design variables for the benchmark optimization problem

Variable description	Symbol and domain	Step size
South, east, west, north window upper positions (m)	$x_1, x_2, x_3, x_4 \in [1.35,2.7]$	0.01
Building orientation (°)	$x_5 \in [0,180]$	1
Thermal conductivity of opaque envelope (W/m·K)	$x_6 \in [0.008437, 0.1534]$	0.001

Settings of optimization algorithms

Three commonly used algorithms, namely, Hooke–Jeeves algorithm, Multi-Objective Genetic Algorithm II (MOGA-II), and Multi-Objective Particle Swarm Optimization (MOPSO) algorithm are compared in this study. The Hooke-Jeeves belongs to the family of direct search algorithms and is implemented in GenOpt. It starts from a single point in the design space and then performs local search and pattern search alternatively. Both MOGA-II and MOPSO are stochastic and population-based algorithms and are applied in modeFRONTIER. The MOGA-II is inspired from the biological principles of natural selection and genetic recombination, while the MOPSO mimics the social behavior of flocks of birds or schools of fish.

For the control parameters of the Hooke–Jeeves algorithm, this study uses a mesh size divider of 2, an initial mesh size exponent of 0, a mesh size exponent increment of 1, and the maximum number of step size reductions of 4 (Anon. B). The main parameters used in the MOGA-II are the directional cross-over, selection, and mutation probabilities. Their default values employed in modeFRONTIER are 0.5, 0.05, and 0.1, respectively (Anon. C). For the MOPSO algorithm, default parameter settings include a turbulence of 0.2 and a random generator seed of 1.

RESULTS AND DISCUSSION

Table 2. Results of the 18 optimization runs driven by the 3 algorithms with the set of algorithm control parameters and initial design of experiment unchanged

	Index	Number of Simulations	x_1 (m)	x_2 (m)	x_3 (m)	x_4 (m)	x_5 (°)	x_6 (W/m·K)	Objective (KW·h/m²·a)	$d(X^*, X')$	$g(f(X^*), f(X'))$ (%)
Initial solution	-	-	1.63	1.80	2.66	1.35	132	0.067422	157.83	1.3293	10.61
True optimal solution	-	-	1.35	1.35	1.35	1.35	4	0.008437	142.692	0	0
Hooke–Jeeves	1-6	256	1.4	1.35	1.4	1.35	180	0.008485	142.98	0.9792	0.2021
MOGA-II	1	236	1.35	1.35	1.35	1.35	178	0.008437	142.712	0.9667	0.014
	2	298	1.35	1.35	1.35	1.35	159	0.008437	142.802		
	3	292	1.35	1.35	1.35	1.35	2	0.008437	142.693		
	4	295	1.35	1.35	1.35	1.35	4	0.008437	142.692		
	5	256	1.35	1.35	1.35	1.35	180	0.008437	142.704		
	6	292	1.35	1.35	1.35	1.35	162	0.008437	142.788		
MOPSO	1-6	270	1.35	1.35	1.35	1.35	0	0.008437	142.696	0.0222	0.008

For each of the three algorithms, six optimization runs are repeated with the same initial solution and algorithm control parameter settings. Table 2 shows the optimal solution searched in each optimization run and their closeness to the true optimal solution.

Results shows that six optimization runs derived by the Hooke–Jeeves algorithm and the MOPSO algorithm yield the same results, that is, the search processes and optimal solutions are identical. Thus, the stability of both algorithms is very perfect. For the MOGA-II, although optimal solutions searched by six runs are not the same, there is little difference between their objective function values, with standard deviation being only 0.05 $KW{\cdot}h/m^2{\cdot}a$. Thus, the MOGA-II also performs good in terms of stability.

Taking the first optimization run driven by each algorithm to compare their performance in validity, speed and coverage. As Table 2 shows, the values of $d(X^*, X')$ and $g(f(X^*), f(X'))$ of the MOPSO algorithm are the smallest among the three algorithms, indicating that the optimal solution found by the MOPSO algorithm is closest to the true optimum and its validity is the highest. The $g(f(X^*), f(X'))$ value of the MOGA-II is small and similar to that of the MOPSO algorithm, whereas its $d(X^*, X')$ value is quite large. The Hooke–Jeeves algorithm obviously does not find a satisfactory solution because the values of both $d(X^*, X')$ and $g(f(X^*), f(X'))$ are the largest. Therefore, the Hooke–Jeeves algorithm has the worst performance in validity compared with the MOPSO algorithm and the MOGA-II.

As shown in Table 2, the MOGA-II finds its optimal solution with the fewest simulation number of 236, the Hooke–Jeeves algorithm needs 256 simulations, and the MOPSO algorithm requires 270 ones. Therefore, the MOGA-II performs quite better than the other two algorithms in terms of speed.

Table 3. *Standard deviations of all searched solutions by each algorithm in each variable domain and their products*

	Standard Deviation						Product
	x_1	x_2	x_3	x_4	x_5	x_6	
Hooke-Jeeves	0.081	0.143	0.437	0.003	15.21	0.019	4.3884E-06
MOGA-II	0.212	0.261	0.173	0.404	39.791	0.018	2.76994E-03
MOPSO	0.351	0.338	0.339	0.343	52.563	0.035	2.5378 E-02

To quantify and compare the coverage of the three algorithms, Table 3 calculates and lists the standard deviations of all solutions searched by each algorithm in each variable domain and their products. The table reveals that the standard deviations of the MOPSO algorithm are generally larger than those of the MOGA-II. The product value of the standard deviations in the six variable domains of the MOPSO algorithm is an order of magnitude larger than that of the MOGA-II and four orders of magnitude larger than that of the Hooke–Jeeves algorithm. Therefore, the MOPSO algorithm is outstanding in the coverage, indicating excellent capability of searching the entire solution space without being trapped in local optimums. The MOGA-II performs slightly worse than the MOPSO algorithm but is still able to conduct a fairly global search. However, the Hooke–Jeeves algorithm performs poorly in coverage and is vulnerable to local optimum traps.

CONCLUSIONS

The aim of this research is to develop a set of criteria to evaluate the performance of algorithms used in building energy efficient design optimization. To verify this method, experiments are conducted using the DOE benchmark medium office building model, and the performance of three algorithms in solving the standard problem are compared utilizing the criteria proposed. It is concluded that the MOPSO algorithm generally performs better than the other two algorithms except for the speed. The Hooke-Jeeves algorithm tends to fall into the local optimum trap with poor validity and coverage. Particularly, there is no algorithm performs best in all four criteria. So when facing an energy optimization problem, the algorithm should be carefully selected based on the natures of the problem and the performance indices that matter the most.

ACKNOWLEDGEMENTS

This paper is financially supported by the Ministry of Science and Technology of China (project number: 2016YFC0700102).

REFERENCES

Anonymity A http://commercialbuildings.energy.gov/reference_buildings.html, last accessed on 26 April 2016.

Anonymity B http://simulationresearch.lbl.gov/GO/download/manual-3-1-0.pdf, last accessed on 26 April 2016.

Anonymity C http://www.esteco.com/modefrontier/, last accessed on 26 April 2016.

Shi X, Tian Z, Chen W, Si B, Jin X. 2016. A review on building energy efficient design optimization from the perspective of architects. *Renewable and Sustainable Energy Reviews.* 65:872-84.

Terzidis K. 2006. *Algorithmic architecture*. Elsevier and Architectural Press.

Nguyen AT, Reiter S, Rigo P. 2014. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy.* 113: 1043-58.

Liang JJ, Qu BY, Suganthan PN, Hernández-Díaz AG. 2013. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. Final Report for Technical Report, Nanyang Technological University (Singapore).

Tuhus-Dubrow D, Krarti M. 2009. Comparative Analysis of Optimization Approaches to Design Building Envelope for Residential Buildings. *ASHRAE Transactions*. 115(2).

Futrell BJ, Ozelkan EC, Brentrup D. 2015. Optimizing complex building design for annual daylighting performance and evaluation of optimization algorithms. *Energy and Buildings*. 92: 234-45.