BUILDING ENERGY OPTIMIZATION USING BUTTERFLY OPTIMIZATION ALGORITHM (BOA)

Mehdi GHALAMBAZ¹, Reza JALILZADEH YENGEJE ² *, Amir Hossein DAVAMI ²

¹ Department of Environmental Engineering, Ahvaz Branch, Islamic Azad University, Ahvaz, Iran
² Department of Environmental Management (HSE), Ahvaz Branch, Islamic Azad University, Ahvaz

* Corresponding author; E-mail: rezajalilzadeh@memail.com

The Butterfly Optimization Algorithm (BOA) is a novel meta-heuristic optimization algorithm, inspired by the intelligence foraging performance of butterflies. In the current research, BOA was employed to minimize the energy consumption of an office building in Seattle. The annual energy demand of the building was computed by EnergyPlus (EP) software. A two-way coupling was established between EP and BOA. The EP takes into account the non-linear interaction of design variables and computes the energy demand of the building. Then the computed amount of energy demand would be transferred to the BOA, where the optimization algorithm decides about changing the design variables. Then, a new set of design variables would be transferred to EP for a new simulation. Through the dynamic interaction of BOA and EP, a building with minimum energy demand was designed. The impact of the number of butterflies on the performance of the optimization algorithm was also investigated. It was found that using 50 butterflies would lead to the best optimization performance. A comparison between the present method and literature optimization methods was made, which showed that BOA with 15 butterflies or higher could adequately avoid local minimums and reach the best minimum with a reasonable computation effort.

Key words: Building optimization problems (BOPs); Butterfly Optimization Algorithm (BOA); building energy demand; optimum building design

1. Instructions

The buildings consume about 40% of the world energy and produce about 36% of the global CO2 [1, 2], and this rate has an increasing trend [1]. Around 57% of the energy demand of constructions is related to air conditioning and lighting purposes [3]. As a result, clearly reducing the energy demand of constructions is an essential task [4, 5].

A building optimization is a practical approach that can minimize the energy demand of a building design systematically. However, the energy demand of a building is a function of weather profile, usage profile, geographical location, and constructing materials. The variation of each design parameter could induce non-linear impacts on the other parameters. Thus, the building optimization issue is a complex and non-linear problem. The energy demand of a building could be computed through numerical simulations. The numerical simulations demand high computational cost and
iterative solution of algebraic equations, and they can estimate the energy demand of a building. The energy consumption of a building is related to its design parameters, which most of the time are adjustable. For example, the thickness of exterior walls, the size of windows, the hangings, and other parameters could be easily adjusted during a design. The literature review demonstrates that the variation of design parameters could effectively reduce the building energy demand [2, 5-7]. Thus, Building Optimization Problems (BOPs) have attracted the attention of many researchers in recent years with the aim of minimization of building energy consumptions [8-10]. The intelligence approaches have also been used to model the buildings [11].

The Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) methods are well-investigated approaches for BOPs. These approaches do not require function gradients, and they can effectively avoid local optimums. These advanced approaches apply systematic search strategies, and hence, they require many building simulations, and their convergence rate is slow. Each building simulation is a time-consuming step with high computational cost, and hence, some of the BOPs could take months [12]. Moreover, since the BOPs are non-linear problems, some of the optimization approaches may be caught in a local extremum and fail to find the global optimum [13]. Hence, new optimization methods with new capabilities are highly demanded to deal with BOPs.

Michalek et al. [14] employed GA, Sequential Quadratic Programming (SQP), and Simulated Annealing (SA) methods to minimize the energy consumption of a building design. Moreover, the ant colony optimization method was applied to seek a trade-off between the cost of a media center in Paris and the lighting performance [15]. Ilbeigi et al. [16] utilized a neural network to learn the energy consumption behavior of an office building. Then GA was used to find the optimum control parameters of the building. The energy consumption of the office could be reduced by 35%, using the optimization approach.

Each optimization method has some advantages and disadvantages, and hence, there is no general optimization method that could dominate all of the literature optimization methods [17]. However, some of the optimization methods could show better performance for a specific type of optimization problems. Wetter and Wright [18] surveyed the capability of the Hooke–Jeeves (HJ) and GA algorithms for reducing the energy consumption of buildings. These authors showed that GA could reach an optimum design with a fair computational cost while HJ had a high possibility of local minimum entrapping. Zhou et al. [19] applied several optimization methods to BOPs. They employed EnergyPlus as the simulation software and a module for optimization. The Nelder Mead Simplex (NMS), GA, SA, quasi-Newton, and tabu search algorithm were some of the investigated methods. The authors found that NMS could efficiently minimize the energy consumption of an office building.

Wetter and Wright [20] explored the optimization performance of nine optimization methods for BOPs. These results of this investigation indicate that the PSO-HJ could be the best optimization approach with minimum energy consumption. This is while the NMS method has the tendency to being caught in local minima. Other researchers such as Kämpf et al. [21], Bucking et al. [22], Futrell et al. [23], and Hamdy et al. [24] have also examined the optimization methods for BOPs.

In a recent excellent investigation, Waibel et al. [25] published a systematic survey of various optimization approaches for minimization of energy consumption in buildings. They concluded that an optimization method with a high convergence rate could fail to find a global optimum due to the risk of being caught in local minima. Furthermore, no optimization method could be dominantly best for BOPs.
Nowadays, new advanced optimization methods are introduced by researchers. Many of these approaches show dominant performance over the regular methods available in the literature. Thus, the could be promising for BOPs. However, the performance of these new methods has not been tested for BOPs. Thus, the examination of new optimization methods for BOPs is highly demanded.

The novel Butterfly Optimization Algorithm (BOA) was proposed by Arora and Singh [26] for solving global optimization problems. The method mimics the mating and food search behavior of butterflies. Butterflies use their smelling sense to locate a mating partner or nectar site. Thus, The BOA was built following butterflies smelling sense and cooperative behavior in their foraging strategy. In this algorithm, the butterflies emit some fragrance, enabling them to communicate with other butterflies.

Although BOA was proposed just in 2019, it has been employed for the optimization of many scientific and engineering problems in a short time. For instance, Yıldız et al. [27] applied BOA in automobile design. This algorithm was employed to find an optimized shape for a suspension arm of a vehicle. The BOA successfully reduced the weight of the component by 32.9%. Tan et al. [28] applied the BOA method to solve partial differential equations (PDEs). The authors used the optimization method to find the coefficients of a complex non-linear function in order to satisfy the partial differential equation. BOA has also been used for power generation optimization of photovoltaic arrays by control of shadings [29], and regression analysis for software testing [30]. BOA is an attractive method that does not require gradients of the objective function, and it contains only a few setting parameters. The algorithm has a fair balance between exploitation and exploration during search phases.

As mentioned, Ilbeigi et al. [16] utilized the combination of a neural network and GA to optimize an office building. Thus, the computations of the EnergyPlus should be saved and then transfer to a neural network. However, in the present study, we developed a coupling interface to perform the communications between the EnergyPlus and the optimization method directly. Moreover, attention to literature works reveals that the BOA was capable of dealing with complex and non-linear optimization problems efficiently. However, this optimization approach was never applied to the buildings and optimization of their energy demand. Thus, the current research aims to apply BAO for minimizing the energy demand of an office building.

2. Methods and materials

The current research aims to examine the optimization capability of BOA in dealing with the minimization of energy consumption in buildings. Here, a benchmark office construction located in Seattle was selected as the test case. Then, the BOA was applied to minimize the annual energy consumption of the office. Thus a Building Energy Optimization (BEO) problem was established in the present research. The BEO approach involves three main blocks. The first block is building simulation and the computation of energy consumption. The optimization method is the second step, which receives the simulation data and controls the decision variables. The third block is a coupling interface that connects the first and second blocks. The details of each block will be described later.

2.1. The building models

Here a benchmark office consist of four design variables was considered [20]. Several researchers analyzed this office building, and here, we took it from [25]. Table 1 shows the name and...
range of design variables, which are the windows sizes for the East and West façades, the shading transmittance, and the orientation of the construction. The details of wall materials, ceiling, floor, and windows have been described in [20, 25], and the office model is also available in supplementary materials. Thus, the details were not repeated here for the sake of brevity.

The annual energy consumption per unit of the floor for the office was introduced the same way as [20, 25] by:

\[
F(X) = \left( \frac{Q_h(X)}{\eta_h} + \frac{Q_c(X)}{\eta_c} + (PEF) E(X) \right) / A
\]

(1)

In the above equation, \(Q_h(\cdot), Q_c(\cdot), \) and \(E(\cdot)\) represent the annual (kWh/a) cooling, heating, and lighting energy demand of office, respectively. Electricity consumption was multiplied by a primary energy factor \((PEF)\) of 3.0, while the cooling \((\eta_c = 0.77)\) and heating \((\eta_h = 0.44)\) efficiencies were applied. Thus, Eq. (1) shows the annual energy consumption of the office per unit area and should be minimized. This equation denotes the objective function (OBJ) for the optimization algorithm. Fig. 1 illustrates a 3D model of the office, which was designed in EnergyPlus (EP). The EP model of the office with all geometrical details and utilized materials has been included in the supplementary materials.

Fig. 1: The model of office building introduced in EnergyPlus is identical to the model of Wetter and Wright et al. [20].

Table 1: The control parameters of the office building

<table>
<thead>
<tr>
<th>Control parameter</th>
<th>Description</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>orientation of the building</td>
<td>([-180, 180]) deg</td>
</tr>
<tr>
<td>(X_2)</td>
<td>width of the west window</td>
<td>([0.1, 5.9]) m</td>
</tr>
<tr>
<td>(X_3)</td>
<td>width of the east window</td>
<td>([0.1, 5.9]) m</td>
</tr>
<tr>
<td>(X_4)</td>
<td>shading transmittance</td>
<td>([0.2, 0.8])</td>
</tr>
</tbody>
</table>
3. Modeling solution

3.1. EnergyPlus for building energy simulation

In the present research, EnergyPlus was used to simulate the energy consumption in the office and compute the annual energy consumption. The EP has been developed by the US Department of Energy [31], and its capability and accuracy in building energy simulations have been tested in the literature. The computational core of EP involves the legacy programs of DOE-2 and BLAST [31], so the computational cost of EP is fair with high accuracy. EP solves the following implicit finite difference scheme for conservation of energy in building elements [32]:

\[
C_p \rho \Delta x \frac{T_{i,j+1}^{t+1} - T_i^t}{\Delta t} = \frac{1}{2} \left( k_w \frac{T_{i,j+1}^{t+1} - T_{i+1,j+1}^{t+1}}{\Delta x} + k_e \frac{T_{i+1,j+1}^{t+1} - T_{i,j+1}^{t+1}}{\Delta x} + k_w \frac{T_{i+1,j}^t - T_{i,j+1}^t + T_{i,j}^t - T_{i,j}^{t+1}}{\Delta x} \right)
\]  

(2)

where \( T \) is the node temperature, \( \Delta x \) is the finite difference layer thickness, \( \Delta t \) is the calculation time step, while \( j \) is the previous time step. Here, \( i \) is the node being modeled, \( i+1 \) is the neighbor node to the exterior of construction \( j \), and \( i+1 \) is the neighbor node to the interior of the construction. \( C_p \) and \( \rho \) represent the specific heat of the material, and density, respectively. \( k_w \) shows the thermal conductivity for an interface between \( i \) node and \( i+1 \) node, and \( k_e \) is the thermal conductivity for the interface between \( i \) node and \( i+1 \) node.

The model of the office can be introduced in EP by using an idf file containing the geometry and material properties of the building. The idf files are indeed structured text files. EP gets an idf file of a building along with the weather-profile and simulated the energy consumption for the model. After computations, it writes the outcomes in text files. The data in the text files can be retrieved by a text editor or an in-house code and evaluate the OBJ function.

3.2. Butterfly Optimization Algorithm (BOA)

The Butterfly Optimization Algorithm (BOA) is a nature-inspired optimization algorithm, which has been recently introduced by Arora and Satvir [26] in 2019. The BOA was inspired by the foraging and mating strategy of butterflies in nature. The following three rules govern the BOA algorithm: 1. The butterflies propagate some fragrance, and hence, they can attract each other; 2. Every butterfly could move randomly or to the finest butterfly propagating a higher degree of fragrance. 3. The incentive strength of a butterfly is under the influence of the foraging site or objective function.

Implementation of BOA involves three phases, which are the initial, iteration, and final phases. At the beginning of the optimization, BOA first executes an initialization subroutine and creates some initial butterflies (\( N_B \)), which represent the solution space and the computed objective function. The number of butterflies will remain constant during the optimization.

Then the butterflies start searching for the optimum solution during an iteration process. In each iteration or a generation (\( N_G \)), the butterflies change their location in the solution space. Thus, their fitness value (OBJ function) should be recomputed. Following the fitness value, butterflies will generate fragrance. Then the butterflies will move toward the best butterfly while they are searching around their location. At the final phase, BOA will stop when it reaches an optimum solution. Since the objective function should be evaluated for all butterflies (\( N_B \)) at each generation (\( N_G \)) the total OBJ function evaluation will be \( N_B \times N_G \). Details about the movement equations of the butterflies, fragrance...
propagation, and mathematics of BOA could be found in [26]. The source codes of BOA are also provided in the supplementary files of the paper. Here the original source codes were adopted from [26], and then it was modified in the form of a function to be included in the coupling interface. The modified version of BOA has been included in supplementary files.

3.3. Coupling BOA and EP

The EP can simulate the energy consumption in the office building, and the BOA has the ability to search for the optimum solution for an OBJ function. However, the BOA is an independent code which needs to receive the value of the OBJ function at each generation, while EP is another independent software, which needs to receive the values of design parameters to compute the OBJ function. Thus, a Coupling Interface (CI) is essential to connect the BOA and EP in both-ways communication. Fig. 2 shows the framework of the communication between BOA and EP, which should be handled by the coupling interface. Fig. 3 depicts a schematic view of the optimization process. Here, we wrote an in-house code to read the idf files and inject the values of design parameters in the idf model. Then, CI should execute EP along with the weather file and wait until EP complete the computations and write the energy consumption data into output files. Then, CI should read the output files and compute the OBJ function using Eq. (1). At this stage, the OBJ function value will be sent to BOA using CI. BOA will compute the butterfly movements and update the design parameters. The design parameters will be received by CI where CI will inject them to the idf file, and the loop will continue until an optimum solution reaches. The final design parameters will be reported as the optimum solution, which minimizes the annual building energy consumption.

![Diagram of the optimization framework consisting of the EnergyPlus and BOA code.](image)

**Fig. 2:** The optimization framework consisting of the EnergyPlus and BOA code.
Fig. 3: The conceptual model of the coupling between BOA and EP.

3.4. Replication of Results

All of the codes and models will be published along with paper, which can be accessed from here: (https://data.mendeley.com/datasets/xtzkjkgr/draft?a=e2b1cd12-fd7d-4387-b309-d68c7941512c) and after publication with DOI: 10.17632/xtzkjkgr.1 (a reserved DOI).

4. 4. Results and discussions

The integration of BOA, CI, and EP was used to minimize the annual energy consumption of the office building introduce in Section 2. This office was also investigated by [25], and they adopted a fixed computational budget of 500 OBJ function evaluations. We also took the same budget of 500 evaluations, so the computational cost of simulations will be the same as the literature-works, and consequently, the comparison between the optimization results will be fair. As mentioned, each butterfly at each generation requires an OBJ function evaluation. Hence, by assuming a constant value of 500 evaluations, the number of generations could be considered as \( N_G = 500/N_p \). It should be noted that the initial butterflies are created based on a random distribution on the search space. Thus, the final convergence and performance could be affected by the initial phase. Here we repeated the optimization 20 times so the results could be plotting in boxplot format.

Fig. 4 shows the effect of the number of butterflies on the performance of the BOA for minimization of the office building energy consumption. Since the optimization process was repeated 20 times for each setting, the results of each case were plotted in the statistical way of using boxplots. For each boxplot, the bottom line shows the minimum value of F(X), while the top lines show the maximum evaluated value of the objective function (energy consumption). The middle lines show the median of the obtained energy consumption for repeated computations. The bottom and the top of the boxes denote the first quartile and the third quartile, respectively.
Fig. 4: The impact of the number of butterflies on the evaluated minimum energy consumption (F(X)). The boxplots show the results of 20 repeated computations for each case.

As seen, using five and ten butterflies as the initial population could lead to high median values and large maximums. Thus, the chance of being caught in local optima will be high. Using 15 and more butterflies give a fair response since the butterflies could effectively search the solution space. Fig. 4 shows that using 50 butterflies could give the best median and also minimum and maximum range of responses. Thus, the case of five butterflies was selected as the worst setting for BOA, while the case of 50 butterflies was adopted as the best setting.

The details of the 20 runs for the cases of five and 50 butterflies are summarized and reported in Tabs. 2 and 3. Moreover, the obtained values of the control parameters are also reported in the tables. For five butterflies, it is interesting that the second row has found the minimum value of F(X)=132.995. However, the best-obtained value for the case of 50 butterflies was corresponding to the 20th row with a minimum value of F(X)=133.544. Attention to the evaluated control parameters of Tab. 2 shows that the optimum values of X_1 have been scattered around the negative values of 73 and the positive value of 71. Thus, a fair number of butterflies are required to search the solution space effectively.

Table 2: Summary of the 20 runs for the evaluated minimum energy consumption and the corresponding optimum control parameters and computed optimized control parameter for the case of 50 butterflies

<table>
<thead>
<tr>
<th>Iteration</th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>X_4</th>
<th>F(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.0313</td>
<td>5.9</td>
<td>5.0906</td>
<td>0.3015</td>
<td>133.7106</td>
</tr>
<tr>
<td>2</td>
<td>72.8036</td>
<td>4.4463</td>
<td>5.0434</td>
<td>0.4478</td>
<td>135.5902</td>
</tr>
<tr>
<td>3</td>
<td>41.0096</td>
<td>5.1317</td>
<td>4.7833</td>
<td>0.3163</td>
<td>136.0182</td>
</tr>
<tr>
<td>4</td>
<td>-79.557</td>
<td>5.0035</td>
<td>5.9</td>
<td>0.2253</td>
<td>135.045</td>
</tr>
<tr>
<td>5</td>
<td>55.2221</td>
<td>4.5243</td>
<td>5.9</td>
<td>0.3081</td>
<td>135.0829</td>
</tr>
<tr>
<td>6</td>
<td>117.9513</td>
<td>5.2985</td>
<td>5.2293</td>
<td>0.3383</td>
<td>134.8482</td>
</tr>
<tr>
<td>7</td>
<td>68.0239</td>
<td>5.2554</td>
<td>5.2074</td>
<td>0.4543</td>
<td>134.3381</td>
</tr>
<tr>
<td>8</td>
<td>-91.421</td>
<td>5.423</td>
<td>5.2057</td>
<td>0.3059</td>
<td>134.9864</td>
</tr>
<tr>
<td>9</td>
<td>-67.7456</td>
<td>5.513</td>
<td>5.9</td>
<td>0.2</td>
<td>134.4673</td>
</tr>
</tbody>
</table>

Fig. 5(a) illustrates the computed energy consumption of the office for the best and worth settings of BOA. The results are plotted against the literature investigations for various optimization problems of the same office building, as discussed by [25]. This figure shows that the badly tuned settings of BOA, i.e., five butterflies (Worst BOA), could produce out of range results. This is while the fine-tuned case of 50 butterflies (Best BOA) leads to fair results. Thus, using the correct number of butterflies to explore the search space is an essential task for the application of BOAs in building optimization.

Fig. 5: (a) The boxplots of literature studies for building energy minimization using various optimization approaches [25], and the results of the BOA. The boxplots are plotted for five butterflies (worst-case) and 50 butterflies (the best case), and (b) The convergence history of BOA based on the generation for various butterflies.

Fig. 5(b) shows the energy consumption at various generations of the BOA for different initial butterflies. Since the total function evaluations were fixed at 500 evaluations, the increase of initial butterflies reduces the number of possible generations to cope with the function evaluation limit. As seen, the case of five butterflies generally could not find the global optimum solution. However, using 25 butterflies could find a fair optimum point with few generation evaluations. The case of using 50 butterflies not only find a good initial solution but also quickly drop the objective function to the best minimum energy consumption. This is an interesting advantage of BOA, which tends to reach a converged solution quickly with a few generations. Hence, it can be concluded that a BOA with a fair number of butterflies could be applied for BOPs efficiently.
The time history of function evaluations for two cases of five butterflies (worst setting) and 50 butterflies (best setting) has been plotted in Figs. 6 (a) and (b), respectively. In both cases, there are butterflies that tend to find the best minimum energy consumption (minimum values) while there are also other butterflies that scouting for possible better solutions. In the case of 5 butterflies (Fig. 6a), the majority of function evaluations, F(X), are at the bottom of the figure, which shows that only a few butterflies were searching for a better optimum point while the others were following the best local optimum. In contrast, Fig. 6b shows a fair distribution between the minimum points and maximum points. This means that there were a fair number of butterflies that could efficiently search the domain of solution for global optimum and help BOA to scape a local minimum.

![Fig. 6: The convergence history of BOA based on the function evaluations for (a) five butterflies, and (b) 50 butterflies.](image)

5.5. Conclusion

The butterfly optimization algorithm was coupled with the EnergyPlus software using and in-house coupling interface. Then the combination of BOA, EP, and CI was used to minimize the annual energy consumption of an office building. The influence of BOA, the number of butterflies, was surveyed on the minimization performance of BOA. The outcomes reveal that recurring 50 butterflies could result in the highest performance and fastest convergence while requiring five butterflies would significantly decline the BOA performance due to lack of exploration. The optimizations were repeated 20 times for each setting parameters, and the results were plotted as boxplots. The boxplots of the best and worst BOA settings were plotted along with the literature methods for a fixed computational budget of 500 OBJ function evaluations.

The results show that using 15 butterflies or more could result in a reasonable optimum solution. A comparison of the results with the literature outcomes shows that BOA with 50 butterflies could provide fair optimum outcomes with a fixed computational budget. However, the BOA with few butterflies had a high tendency to be entrapped in local optima and fail to reach an excellent optimum solution.

The search history of BOA in the form of optimum solution per generation and history of function evaluations demonstrate that BOA has a very high convergence rate, and hence, it could be much advantageous in the tasks with low computational budgets. For example, attention to the
function generation results shows that BOA could reach almost the same results with a few generations, and continuing the generations could not reduce the solution notably.

Nomenclature

\( A \) area (m\(^2\))
\( T \) node temperature (K)
\( C_p \) specific heat capacity (J Kg\(^{-1}\) K\(^{-1}\))
\( PEFI \) Energy factor (-)
\( E(X) \) the annual lighting energy (kWh/a)
\( Q_h \) heating energy (kWh/a)
\( F(X) \) the annual energy (kWh/a)
\( Q_c \) cooling energy (kWh/a)
\( k \) the thermal conductivity (W m\(^{-1}\)K\(^{-1}\))
\( X \) optimization variable (m, deg)

Greek symbols

\( \rho \) density (kg m\(^{-3}\))
\( \Delta x \) the finite difference layer thickness (m)
\( \eta_c \) cooling efficiency
\( \Delta t \) the calculation time step (s)
\( \eta_h \) heating efficiency

Subscript

\( i \) the node being modeled (-)
\( e \) an interface between \( i \) node and \( i-1 \) node
\( w \) an interface between \( i \) node and \( i+1 \) node

Superscript

\( j \) the previous time step (-)

Acknowledgment

This study was partially supported by Ahvaz Branch of Islamic Azad University, and the authors would like to thank the Research Council for their generous support regarding this research.

References


