RESEARCH ARTICLE

Using artificial neural network and WebGL to algorithmically optimize window wall ratios of high-rise office buildings

Shenghuan Zhao 1,2,*

1School of Architecture and Urban Planning, Suzhou University of Science and Technology, Suzhou 215011, China and 2Jiangsu Key Laboratory of Intelligent Building Energy Efficiency, Suzhou 215009, China

*Corresponding author. E-mail: shenghuan.z@gmail.com http://orcid.org/0000-0003-3109-6087

Abstract

By coupling parametric modeling, building performance simulation engines, and optimization algorithms, optimal design choices regarding predefined building performance objectives can be automatically obtained. This becomes an emerging research topic among scholars in the fields of architecture and built environment. However, it is not easy to apply this method to real building design projects, because of two main drawbacks: Building performance simulation is too time consuming, and the numerical visualization of final results is not intuitive for architects to make decisions. Therefore, this study tries to fill these two gaps by training an artificial neural network to replace simulation engines and developing a web application to speed up the 3D visualization of selected design choices. These two strategies are applied to optimize office towers’ window wall ratios in Hangzhou, China. Architects working on new design projects in that city can obtain the optimal group of window wall ratios for four facades in 2 s, faster than using simulation engines, which cost architects 2 weeks. Moreover, architects can also efficiently observe the appearance of design solutions with the web application. By improving its usability from these two aspects, this study significantly improves the applicability of algorithmic optimization for building design projects.

Keywords: building façade design; design automation; algorithm-aided design; building performance simulation; neural network

Abbreviations

ANN: Artificial neural network
ASF: Area of a standard floor
CEUI: Cooling energy use intensity
HEUI: Heating energy use intensity
HVAC heating: Ventilation and air conditioning
NoF: Number of floors
SUDI: Spatial useful daylight illuminance
WWR: Window wall ratio
WWRE: Window wall ratio on the east
WWRS: Window wall ratio on the south
WWRN: Window wall ratio on the north
WWRW: Window wall ratio on the west

1. Introduction

Window wall ratio (WWR), the ratio between the total glazing area and the whole building facade area, plays a crucial role in building performances, like energy consumption and indoor visual comfort (Hiyama & Wen, 2015; Vanhoutteghem et al., 2015; Marino et al., 2017; Xu et al., 2018; Troup et al., 2019). On the other side, WWR also influences the aesthetics of building facades,
because the comparison between the transparent and the solid offers people a strong visual perception of building appearances. Therefore, deciding WWRs is a crucial step to design building facades.

There have been many studies on algorithmically optimizing WWRs (Chiesa et al., 2019; Zhai et al., 2019). They coupled building performance simulation software with optimization algorithms to obtain optimal design choices. However, traditional building performance simulation, which used simulation software, is too time consuming, especially for those that have complex geometries and many thermal zones. Consequently, the optimization process also takes a lot of time, hindering algorithmic optimization from being applied to real design projects.

To overcome this time consumption issue, artificial neural networks (ANNs) that imitate “cause–effect” relationships in a complicated system can be utilized. Similar to biological neural networks’ functions and structure, ANNs are computational models that can carry out tasks by learning from existing examples. After training an ANN with enough presimulated data, it can predict building performance without running simulation software (Marzbani et al., 2016). There also have been studies using ANNs to quickly predicting building performances, like the energy demand (Paudel et al., 2014; Duarte et al., 2017; Paterson et al., 2017; Amasyali & El-gohary, 2018; Singaravel et al., 2018; Xu et al., 2018; Li et al., 2018; Si et al., 2019), life cycle assessment and life cycle cost (Amirhosain & Hammad, 2019), thermal comfort, time ratio (Gou et al., 2018; Si et al., 2019), and daylight (Lorenz et al., 2019). More comprehensive information about the topic can be found in this literature review (Roman et al., 2020). Besides studies solely using ANN to replace building simulation software, there also have been investigations coupling ANN-based performance simulations with optimization algorithms to aid the building design (Magnier & Haghighat, 2010; Gossard et al., 2013; Asadi et al., 2014; Perera et al., 2019; Si et al., 2019).

Nevertheless, most of these studies were conducted only for specific building cases, usually with determined building shapes, sites, surroundings, etc., and to solve unique or particular problems for that building. Their optimization frameworks were not intently designed for broad usage. Therefore, these studies are seldomly applicable and reusable to other design projects. Meanwhile, because these studies are conducted by mechanical engineers, most of them are HVAC (heating, ventilation, and air conditioning) rather than architectural geometric design oriented (Magnier & Haghighat, 2010). As a result, there seems still no study specifically developing ANN-based optimization investigations for WWR design.

Apart from the applicability issue, there is another gap between those optimization studies and the decision-making process in real design projects. Algorithmic optimization usually shows performance-optimal solutions numerically. However, for real design projects, architects do not only take building performance into account, but also consider architectural aesthetics. Taking WWR design as an example, the amount of transparent areas critically influences the aesthetic effect of detailed window geometries. Neither too much nor too less glass area would lead to ideal facade appearances in the later design elaboration phase. Therefore, when optimization algorithms find performance-optimal design solutions, it is still necessary for architects to quickly and intuitively understand how large the transparent area on a façade is. Nevertheless, ANN-supported algorithmic optimizations are usually conducted in nonvisual programming environments, which are impossible to visualize building design choices in 3D. In this stage, using extra 3D modeling software for visualization would interrupt the decision-making process. Therefore, Web3D technologies have started being applied to architecture, engineering, and construction (AEC; Dobos et al., 2018).

This study aims to fill these two gaps introduced above, making simulation-based algorithmic optimization more applicable for architects to decide office buildings’ WWRs. For the first gap, based on the previous research (Zhao, 2020), the author has obtained a universally applicable and reasonable optimization framework for office buildings’ WWRs. The three most important performance indicators should be initially pursued when designing WWRs: minimizing CEUI (cooling energy use intensity), HEUI (heating energy use intensity), and maximizing SUDI (spatial useful daylight illuminance). CEUI and HEUI, respectively, stand for the amount of energy used for annually cooling and heating an area of 1 square meter. SUDI represents the indicator measuring indoor visual comfort. A more detailed introduction and other information about these building performance indicators can be found in the previous publication (Zhao, 2020).

Afterward, rather than using simulation software, an ANN is trained to predict building performance and then coupled with an optimization algorithm. By doing this, future designers who also design high-rise office buildings located in Hangzhou can immediately obtain the optimized WWRs in a very short time, without using simulation software like Eneryplus and Daysim. For the second research gap, to help architects efficiently check the aesthetics of the amount of fenestration area, a webpage application with the Web Graphics Library (WebGL) technology is developed. This webpage application can visualize fenestrated 3D building models in a fast and easy way. Related information about web-visualization in the AEC industry can be found in this literature review (Wagner et al., 2020).

By speeding up the design optimization and easing the design result visualization, this study dramatically improves the decision-making process for WWR design. Although this study is just for finding the optimal WWRs of high-rise office buildings in Hangzhou, China, the same idea and methodology can be applied to other design scenarios, other building types, and other cities in different climates.

Besides this introduction section, the remaining of this paper is structured as follows: Section 2 presents the methodology, including the general framework, training data acquisition, neural network training, algorithmic optimization, and design choice visualization. In Section 3, this design platform is applied to four case studies to test its effectiveness. Moreover, its usability is also evaluated in a design workshop. Finally, Section 4 offers a summary and a critique of the whole study.

2. Methodology

2.1. The general workflow and instrumentation

As illustrated in Fig. 1, the general workflow of this study can be divided into the following five main steps:

1. describing a building with eight variables and using the Latin Hypercube Sampling (LHS) algorithm to generate near-random building cases.
2. using software Energyplus and Daysim to simulate those cases’ performance.
3. training an ANN to predict three category building performances based on data from the previous step.
4. coupling the ANN with an optimization algorithm to optimize WWRs.
5. selecting one design solution from the optimization results and visualizing it to check its aesthetics.

Detailed methodologies of these steps will be introduced in later subsections. Section 2.2 explains how to acquire data for ANN training. After that, Section 2.3 describes the process of ANN training and validation. Section 2.4 informs readers how to couple the ANN with an optimization algorithm. At last, Section 2.5 illustrates the 3D visualization process.

Figure 2 demonstrates the software and computational tools used to set up the instrumentation platform and how the platform can be used by future designers. The building case sampling is conducted in PyCharm (JetBrains s.r.o., 2020), an integrated programming platform especially developed for Python. The building performances are predicted by simulation engines Daysim and Energyplus, using Grasshopper (Wikipedia, 2018) as the front end. Grasshopper is a visual programming extension of the 3D modeling software Rhino.

Although the previous step is conducted in Grasshopper, which has its own machine learning tools (Msiller, 2018; Greco, 2019), the current ANN training is not executed in Grasshopper due to three reasons. First, according to the author’s experiences, both the ANN training and algorithmic optimization are easy to crash in Grasshopper. Second, ANN training plugins in Grasshopper were developed to increase the usability for architects, but these conveniences are at the cost of flexibilities. For example, in Grasshopper, activation function alternatives for ANN training are very limited and outdated. Third, this study’s productions are planned to be fully web-accessed and cloud-based in the future, but ANNs trained in Grasshopper do not support this objective. Therefore, the ANN training herein is still conducted in PyCharm. To make the ANN easily trained, the ANN training structure is Pytorch (Paszke et al., 2019), based on the Torch library.

In the end, optimal design choices are subjectively selected and perceptually visualized in a newly developed web application. WebGL is a JavaScript API allowing users to create 3D graphics in a web browser. Built on top of WebGL, Three.js is a cross-browser library that can more easily create and display graphics. Using Three.js in the JavaScript IDE Webstorm, the
Algorithmically optimize window wall ratios by using ANN

Figure 2: Software and plugins utilized in this study.

Table 1: Ranges and steps of input variables.

| Parameters | Ranges      | Steps | Number of alternatives |
|------------|-------------|-------|------------------------|
| X₁         | ASF         | [650, 2000] | 1 m²                  | 1351       |
| X₂         | The aspect ratio of the standard floor plan | [1, 4] | 0.1 | 40 |
| X₃         | NoF         | [6, 23] | 1 level | 18 |
| X₄         | Rotation degrees | [−45, 45] | 1 degree | 91 |
| X₅         | WWRE        | [0.17, 0.70] | 0.01 | 53 |
| X₆         | WWRW        | [0.17, 0.70] | 0.01 | 53 |
| X₇         | WWRS        | [0.17, 0.80] | 0.01 | 63 |
| X₈         | WWRN        | [0.17, 0.80] | 0.01 | 63 |

The author creates an HTML file as the web application to visualize selected design solutions from the previous step. More detailed procedures will be presented in the following subsections.

2.2. ANN-training-dataset acquisition

The process of data collection for ANN training includes two steps. The first step is to reasonably parametrize high-rise office buildings into variables and sample a certain amount of building cases to represent the whole design space. The second step is to simulate these building samples and obtain data about their performance. The geometric data will be fed into the ANN’s input layer, and the performance data will be fed into the output layer.

2.2.1. Generate building samples

Because most high-rise office buildings around the world are designed with a regular box-like shape rather than irregular ones, this study only considers typical office buildings. Based on other researchers’ survey about the performance of high-rise office buildings (Raji et al., 2017) and the current author’s working experiences as an architect, one office tower herein is defined by eight geometric variables: ASF (area of a standard floor), aspect ratio, NoF (number of floors), rotation degrees, WWRE (window wall ratio on the east), WWRS (window wall ratio on the south), WWRN (window wall ratio on the north), and WWRW (window wall ratio on the west). The former four variables define the basic building geometry, and the latter four are WWRs on four orientations. Buildings are all located in Hangzhou, China (30.2741° N, 120.1551° E), in the hot-summer and cold-winter climate zone, complex but academically valuable to investigate.

According to design codes in China and architects’ practical experiences, each variable is assigned with a respective value range and step of variable values (Table 1). The first decision variable is ASF. If the ASF of a high-rise building is smaller than 650 m², the building’s vertical traffic core, including emergency stairs and elevators, relatively occupies too much area on one floor, compared with the other usable space. The low floor area utilization rate makes the project economically unreasonable. On the other side, according to the Chinese code of building fire protection and prevention, to provide fire safety, if its ASF is larger than 2000 m², a building needs another sprinkler fire-extinguishing system. Therefore, the ASF of most high-rise buildings in China varies from 650 m² to 2000 m².
The aspect ratio of building floor plans is an essential factor in building energy efficiency (McKeen & Fung, 2014). Although there is no Chinese design code strictly demanding this value range, the maximum aspect ratio is set to 4 here. The reason is that considering the maximum allowable distance between the outmost point of the corridor and the emergency stair, the shape of a floor plan should neither be too thin nor too long.

The minimum interior net floor height required in the Chinese office design code (JGJ67–2006) is 2.7 m. Considering the heights of beams and technical pipes, the average story height is assumed to be 4.2 m in this study. Therefore, the NoF is set from 6 to 23 to limit the building height below 100 m. Otherwise, buildings would belong to super high-rise buildings, which requires a stricter construction code and causes higher investment costs.

Most high-rise offices in China are ordinary high-rise buildings, not super high-rise ones (higher than 100 m).

Speaking of building orientations, buildings in the northern hemisphere are unlikely facing the due west or due east, which causes glare risks and exposure to the intensive heat of the afternoon sun. The range of rotation degrees is set from $-45$ degrees to 45 degrees. If rotation degrees are positive, a building is rotated to face southeast. Otherwise, a building is turned clockwise to face southwest.

The latter four variables are four facades’ WWRs, which are going to be optimized. According to Hangzhou’s local design standard for building energy efficiency, WWRE and WWRW should not be larger than 0.7, while WWRS and WWRN should not be larger than 0.8. All four WWRs are assigned with a step
Algorithmically optimize window wall ratios by using ANN

Figure 4: The Grasshopper code transforming data into building geometries (a); breps of one building visualized in Rhino (b); a fenestrated building with three selected floors for simulation (c).

What has been discussed above is the whole design space, used to represent all possible rough designs for high-rise office towers in Hangzhou, in regular shapes. After that, from this space, the author needs to sample a certain number of cases to createthedatasetforANNtraining. Speaking of how many samples should be retrieved to represent the whole design space, using too few cases will not be able to train an ANN. However, when there are too many samples, it would cost too much time to use software to simulate them in the next step. Studies (Asadi et al., 2014; Siet al., 2019) mentioned that it is plentiful to sample the search space when the number of samples retrieved is more than twice the number of input variables. According to that theory, it would be enough only to use 16 cases in our study (8 times 2). However, it seems that relatively more samples can be simulated in this study. Therefore, based on the author’s former ANN training experiences, this study samples 650 cases from the whole design space.

Building cases herein are sampled from the total design space by the LHS algorithm, a statistical approach that can generate near-random samples from a multidimensional distribution. This algorithm has been frequently used in related studies (Magnier & Haghighat, 2010; Asadi et al., 2014; Ascione et al., 2017; Gou et al., 2018; Siet al., 2019). This study plots obtained samples to check their representativeness. The scatter plot matrix (Fig. 3) identifies that 650 samples obtained have great
Figure 5: The Grasshopper code of performance simulation and data exportation.

diversities in any design space formed by two variables. The diagonal ax shows the univariate distribution of the data for the variable in that column. It confirms that these 650 cases can represent the whole design space, which has been introduced in Table 1. Meanwhile, contents inside the red triangle are what will be optimized. Readers can compare the WWRs here and WWRs after optimization in later sections.

Afterward, with the code shown in Fig. 3a, the dataset containing sampled cases' information is imported in Grasshopper. To make the code short and precise, variables of “aspect ratio” and “ASF” are transformed into “floorplanwidth” and “floorplan depth” in advance. Values of four basic geometric variables are turned into geometries of a building case (Fig. 4b). Afterward, the building is fenestrated with four WWRs (Fig. 4c). By scaling down wall shapes based on given WWRs, one huge window is obtained and positioned on each wall’s centroid. Building performances will be simulated with this kind of fenestration geometries. When the parametric modeling software creates building geometries, all façade designs have a single centered window according to the factor of WWR. Although the result may not like real façade designs, it still offers architects useful information to elaborate detailed ones in the later stage. Moreover, in the previous study (Zhao, 2020), it is confirmed that the distribution of transparent areas does not influence SUDI, CEUI, and HEUI too much.

All these eight variables are indexed by the same slider, connected to an automatic iterator Colibri (Jonatan, 2017). After one building sample is simulated, and its performance information is recorded, Colibri automatically indexes the next group of eight variables, and the new building case starts to be simulated.

### 2.2.2 Building performance simulation

The Honeybee is a series of Grasshopper plugins (Roudsari, 2019), allowing users to execute several building performance simulation software using Grasshopper as the front end. Building case geometries are connected to Honeybee as building thermal zones. Each building’s ground floor, one standard floor in the middle, and the top floor (identified with the brown color in Fig. 3c) are selected to represent the whole building. Average values of these three floors’ performance simulation results are simulated and recorded.

Based on the previous study (Zhao, 2020), when using building performance to drive the façade design, these three building performance indicators are the most critical: SUDI, CEUI, and HEUI. SUDI stands for the degree of indoor visual comfort, while CEUI and HEUI identify the energy efficiency. More specifically, SUDI here identifies percentages of floor area having daylight between “100 and 2000” lux at and above 50% time of the whole year. CEUI and HEUI identify how many kilowatts of energy are required to cool and heat 1 square meter for 1 year. SUDI is simulated by software Daysim, while CEUI and HEUI are simulated by Energyplus. Both simulation engines herein are executed by using the Grasshopper code (Fig. 5) as the front end.

To apply this study to any office tower design in Hangzhou, the selection of construction material should be the most popular kind among all future projects. According to data from the local construction bureau, Extruded Cement Panels and Autoclaved Light-weighted Concrete Panels are emerging selected façade materials in that city. Therefore, detailed material information (Table 2) is retrieved from another publication (Zhao, 2019) that designated these two kinds of materials to a prefabricated wall. The operative temperature of the HVAC system is listed in Table 3, and other schedules are given in Table 4.

The laptop processor used for building performance simulation is Intel® Core™ i7-8550U CPU @1.80GHz-1.99GHz, with 24.0 GB installed memory (RAM). For every single case, Energyplus consumes 1 min to predict CEUI and HEUI, while Daysim spends 10 min accomplishing the SUDI prediction. Due to several unnoticed crashes and the time consumed to restart the program, the whole simulation process for all 650 cases lasted for two weeks. In the end, an Excel file containing 650 groups of eight geometric variables and three performance indicators is prepared to train the ANN. Detailed data of these 650 cases can be found in Appendix 1.
Algorithmically optimize window wall ratios by using ANN

Table 2: The profile of thermal properties in energy models (outdoors to inside).

| Locations                  | Materials                        | Thickness (mm) | Thermal conductivity (W/m·K) | Density (kg/m³) | Heat capacity (kJ/kg·K) |
|----------------------------|----------------------------------|----------------|-----------------------------|-----------------|------------------------|
| External wall              | Extruded cement panel            | 40             | 0.53                        | 75              | 1.05                   |
|                            | Polyurethane                     | 50             | 0.02                        | 30              | 1.05                   |
|                            | Autoclaved lightweight concrete panel | 175          | 0.16                        | 520             | 1.05                   |
| Roof                       | Roof deck                        | 19             | 0.140                       | 530             | 0.900                  |
|                            | Fiberglass quilt                 | 111.8          | 0.040                       | 12              | 0.840                  |
|                            | Plasterboard                     | 10             | 0.160                       | 950             | 0.840                  |
| Normal floor               | Fine aggregate concrete          | 40             | 1.51                        | 2300            | 0.9                    |
|                            | Polyurethane                     | 10             | 0.027                       | 45              | 4.77                   |
|                            | Reinforced concrete              | 120            | 1.74                        | 2500            | 1.05                   |
| Floor exposures to the air | Fine aggregate concrete          | 40             | 1.51                        | 2300            | 0.9                    |
|                            | Polyurethane                     | 10             | 0.027                       | 45              | 4.77                   |
|                            | Reinforced concrete              | 120            | 1.74                        | 2500            | 1.05                   |
|                            | Rockwool panel                   | 30             | 0.045                       | 80              | 0.75                   |
|                            | Fiber-reinforced cement board    | 8              | 0.52                        | 1800            | 1.05                   |

Window properties

- Extinction coefficient: 0.0196/mm
- Number of panes: 2
- Pane thickness: 3.175 mm
- Air-gap thickness: 13 mm
- Index of refraction: 1.526
- Thermal conductivity of glass: 1.06 W/m·K
- Normal direct-beam transmittance through one pane: 0.86
- The conductance of each glass pane: 333 W/m²·K
- Combined radiative and convective coefficient of air gap: 6.297 W/m²·K
- Exterior combined surface coefficient: 21.00 W/m²·K
- Interior combined surface coefficient: 8.29 W/m²·K
- U-value from interior air to ambient air: 3.0 W/m²·K
- Hemispherical infrared emittance of ordinary uncoated glass: 0.9
- Density of the glass: 2500 kg/m³
- Specific heat of the glass: 750 J/kg·K
- Interior shade devices: None
- Double-pane shading coefficient at normal incidence: 0.907
- Double-pane solar heat gain coefficient at normal incidence: 0.789
- Visible transmittance of the glass: 0.913

Table 3: The setpoint temperature schedule of heating and cooling on workdays (°C).

| Hours | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------|---|---|---|---|---|---|---|---|---|----|----|----|
| Cooling | 37 | 37 | 37 | 37 | 37 | 37 | 28 | 28 | 28 | 26 | 26 | 26 |
| Heating | 12 | 12 | 12 | 12 | 12 | 12 | 18 | 18 | 18 | 20 | 20 | 20 |
| Hours | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 20 | 21 | 22 | 23 |
| Cooling | 26 | 26 | 26 | 26 | 26 | 26 | 37 | 37 | 37 | 37 | 37 | 37 |
| Heating | 20 | 20 | 20 | 20 | 20 | 20 | 12 | 12 | 12 | 12 | 12 | 12 |

Table 4. The schedule of occupancy, lighting, and equipment usage on workdays (%).

| Hours | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------|---|---|---|---|---|---|---|---|---|----|----|----|
| Percentages | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 50 | 95 | 95 | 95 | 80 |
| Hours | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| Percentages | 80 | 95 | 95 | 95 | 95 | 30 | 30 | 0 | 0 | 0 | 0 | 0 |

2.3. ANN training and validation

Before training the ANN, it is necessary to design ANN’s architecture properly. The architecture herein is the most basic and common one, the fully connected neural network. It is clear that eight geometric parameters are eight neurons for the input layer, and the results of three performance indicators are three neurons for the output layer. Due to it being a simple dataset, the author sets two hidden layers, layers between the input layer
and the output layer. The number of neuron(s) on each hidden layer is decided after many manual trials of tuning to avoid underfitting and overfitting. Therefore, the current ANN structure contains two hidden layers, and each layer is composed of 512 neurons.

The activation function for every neuron herein, defining the output of one node given an input or set of inputs, is Rectified Linear Unit. This piecewise linear function makes a model more easily trained and achieve better performance.

After that, the author divides the dataset into a training set and a testing set. The quantity ratio between the samples in the training set and in the test set is 4:1. It means that 520 cases are used for training the ANN, and the left 130 cases are used for validating the ANN. The batch size, the number of training examples utilized in one iteration, is set at 32. The learning rate, the amount that the weights are updated during training, is set at $10^{-4}$. The weight decay is set at $10^{-3}$.

The main goal of the ANN learning process is to minimize the loss, which is an aggregation of all errors made for each sample in training or validation sets. The loss function here is ADAM (Kingma & Ba, 2015), the name derived from adaptive moment estimation, making the ANN training converge sooner. For this study, the loss function converges when it comes to 80 epochs.

Figure 6 confirms that the ANN in this study is successfully trained. Figure 6a shows us the progress of training the ANN. As the number of epochs increases, the most suitable coefficients of the loss function are found to make the loss minimized. To further verify the ANN, $R^2$ (R square error), measuring how close the data are to the fitted regression line, is applied to the test dataset. As shown in Fig. 6b-d, $R^2$ values of three kinds of outputs are all above 0.9, identifying the agreement between that the ANN prediction and the software simulation.

2.4. Couple the ANN with the optimization algorithm

After the ANN is trained and validated, it is coupled with an optimization algorithm to search for the optimal WWRs. To avoid the interoperability problem among different software, the evolutionary optimization is also conducted in the Python IDE PyCharm. The optimization tool used here is Geatpy (Geatpy Team, 2019), a genetic and evolutionary algorithm toolbox for Python. It offers many algorithms to choose from, but the one coupled herein is Non-dominated-and-crowding Sorting Genetic Algorithm II (NSGA II), which have been used in several other similar studies (Li et al., 2017; Cascone et al., 2018; Zhai et al., 2019). NSGA II is an effective variation of the basic genetic algorithms (Kheiri, 2018) and is placed as one of the most efficient multi-objective evolution algorithms (Lin & Yang, 2018). It efficiently provides a continuous and iterative improvement of the group of WWRs to reach the goal, by utilizing elitism to strengthen the chance of generating better offspring solutions.

The optimization process is initiated by a group of WWRs randomly selected by the algorithm. Then, these WWRs are fed to the ANN as inputs, and the SUDI, CEUI, and HEUI are predicted by the ANN as outputs, and fed back to the optimization algorithm to go on exploring the design space. As mentioned in the introduction section, the aim of optimization here is to maximize SUDI and minimize CEUI and HEUI. The optimal design solutions found by the algorithm can balance indoor visual comfort and energy consumption for heating and cooling.

After the optimization algorithm NSGA II converges, a Pareto surface composed of nondominated design solutions is obtained. Each point on the Pareto surface stands for one design solution’s performance on three objectives. Users can rotate the
Algorithmically optimize window wall ratios by using ANN

3D Pareto surface in the Matplotlib interface to get a better understanding of optimization results. Geatpy does not keep design solutions that are explored by the algorithm but not optimal. Therefore, the author generates random solutions and plots them in the same diagram with the Pareto surface. Users can check whether these nonoptimal solutions are located on the Pareto surface, furtherly validating the optimization process in a sense. After that, one Pareto surface is projected to 2D spaces to obtain three Pareto frontiers, which identify relationships between two performance objectives.

2.5. Visualize fenestrated buildings

The basic logic to develop this APP is introduced as follows. First, the ASF and aspect ratio are transformed into building width and depth. Second, based on the fixed floor height of 4.2 m, and inputted WWRs, four walls with transparent windows in the middle are built up. After that, the other two faces are created. In the end, the NoF is applied to the model, and the building model is created.

By opening the HTML file with any web browser, inputting eight variables in the interface (Fig. 7), clicking the button to generate, users can visualize the fenestrated building in 1 s. The visualized building is automatically and constantly being rotated, more easily for designers to observe. Moreover, the building model also can be zoomed and rotated manually. When changing inputs and regenerating the building, the previous model information will be erased.

2.6. Basic information of four case studies

WWRs of four building cases with diversely assumed ASF, aspect ratio, NoF, and rotation degrees (Table 5) will be optimized for testing the current ANN-based and WebGL-supported optimization platform. Their optimization results are discussed in the following section.

3. Results and Discussions

This section includes three subsections. In Section 3.1, the optimization results of four building cases are illustrated and discussed. In Section 3.2, among optimal design choices, several of them are selected and visualized in the web application. Afterward, in Section 3.3, architects and architecture students are invited to test this newly developed platform and comment on it.

3.1. Optimization results

One building case herein only costs 2 s to finish the whole optimization process, much faster than using simulation software for the same building case on the same laptop. As already mentioned in Section 2.2.2, if simulation software rather than the ANN is used, it would cost 11 min to simulate one design choice. In total, 2 weeks would be needed to finish the whole optimization process of a similar building case (Zhao, 2020).

The optimal design spaces, 100 optimal design solutions for each case, are plotted in Fig. 8. Each point in these correlograms stands for a group values of design variables \( X_5 \) to \( X_8 \), referring to Table 1 plotted into a 2D dimension, WWRs on four facades. Curves lying on diagonals represent kernel density estimation...
of WWR values. As shown in the figure, it is apparent that each case has a different optimal design space. However, there are still some things in common. For all four cases, the majority of WWRN and WWRW values are below 0.2, in appliance with the Chinese design code of thermal effectiveness. Designing too many transparent areas on these two facades would be energy consuming to heat or cool indoor spaces. For WWRE and WWRS, values vary among the four cases, but all of them have two crests in their curves.

Figure 9 demonstrates the optimization results in terms of building performances. Four Pareto surfaces are shown in Fig. 9a, identifying relationships among the three building performance objectives CEUI, HEUI, and SUDI. Twelve Pareto frontiers, identifying relationships between two objectives, are illustrated in Fig. 9b.

According to those Pareto frontiers, all optimal choices for four building cases have similar HEUI results, from 15 to 20 kWh/m². However, in the perspectives of CEUI and SUDI, design choices for four building cases have various values. When buildings face to the southwest (case 3 and case 4), their design choices bring higher CEUI and smaller SUDI than buildings facing to the southeast (case 1 and case 2). Due to the topic herein being not about built environment science but developing a digital design platform, optimization results will not be furtherly discussed. Detailed data of optimization results can be found in Appendix 2.
Algorithmically optimize window wall ratios by using ANN

Figure 9: Pareto surfaces of four building cases (a); Pareto frontiers of four building cases (b).

3.2. Design choice selection and 3D visualization

Mainly considering the Pareto frontier between CEUI and SUDI, the author chooses one solution from optimal ones for each building case (Table 6). Afterward, to furtherly check the ANN reliability, the author uses traditional software to simulate these four building models’ performances and compare them with results from the ANN. Table 7 tells us that except for the SUDI value of case 1 and the CEUI of case 4, the other indicators predicted by the ANN are very close to results given by simulation software.

Table 6: Selected WWRs for each case and their associated building performances.

| ID  | WWRE | WWRW | WWRN | WWRN |
|-----|------|------|------|------|
| Case 1 | 0.53 | 0.36 | 0.20 | 0.28 |
| Case 2 | 0.20 | 0.18 | 0.63 | 0.18 |
| Case 3 | 0.23 | 0.17 | 0.30 | 0.17 |
| Case 4 | 0.48 | 0.19 | 0.37 | 0.17 |
Table 7: Four building cases’ performances predicted by the ANN and by simulation software.

| ID  | Performance indicators |          |          |          |          |
|-----|------------------------|----------|----------|----------|----------|
|     | SUDI (−)               | CEUI (kWh/m²) | HEUI (kWh/m²) | ANN      | Software | ANN      | Software |
| Case 1 | 91%          | 80.3%     | 93.00    | 94.67    | 18.65    | 17.70    |
| Case 2 | 73%          | 72.1%     | 83.79    | 85.12    | 15.17    | 16.00    |
| Case 3 | 85%          | 86.6%     | 80.12    | 82.25    | 17.92    | 17.62    |
| Case 4 | 79%          | 76.0%     | 86.80    | 112.71   | 16.73    | 17.77    |

In the end, the author visualizes these four fenestrated buildings in the web application (Fig. 10) to obtain subjective feelings towards them: whether the amount of transparent area is aesthetically suitable for the facade. It is an easily reversible process: If an architect does not satisfy with the aesthetic performance of the selected WWR group, considering there are too big or too small transparent areas on some facades, he/she can go back to the previous step and select another design choice.

3.3. Usability test

The effectiveness of this ANN-based and WebGL-supported optimization platform is validated by 50 architects and architecture students in a design workshop. The workshop is divided into three steps: optimization, visualization, and evaluation. First, architects are asked to fenestrate a building case ($X_1 = 1378$, $X_2 = 2.3$, $X_3 = 6$, and $X_4 = 0$, referring to Table 1). After the optimization process is completed, architects select WWRs based on their individual preferences and create the fenestrated building in the web application. In the end, architects evaluate their experiences of using this platform, and additional comments are also encouraged.

Although some architects who do not code need more time to adapt to the Python environment, all 50 participators successfully obtained optimization results (Fig. 11a and b). Meanwhile, Fig. 11c shows how six architects differently chose their preferred design choices. During the investigation, 90% of architects consider that the ANN-based optimization is efficient, and the web application is useful for further window geometry elaboration. The test result identifies that this optimization platform is successfully developed as expected.

4. Conclusions and Future Possibilities

Algorithmic optimization is not easy to be operated in real building design projects, mainly due to the excessive time consumption of building performance simulation. For future high-rise office building design in Hangzhou, the present study speeds up their optimal WWR finding by replacing simulation software with the ANN. Moreover, a web application is created to allow architects to rapidly check the aesthetics of their selected design choices, making the algorithmic optimization more useful. Architects do not need to open heavy modeling software and manually create a geometric model anymore, but only input eight values and click the button.

Figure 10: 3D visualization of selected WWR results of four building cases.
Algorithmically optimize window wall ratios by using ANN

Four case studies are conducted to test this enhanced optimization platform. In the end, the usability test workshop confirms that this new optimization platform offers architects more conveniences, and bridges the gap between emerging digital tools and real design projects.

However, there are also several limits of this study. The main problem exists in Section 2.2.1. Due to this study’s aim being to make a platform for universal use, the specialties of single buildings are sacrificed. For example, the construction materials are defined the same. Although this type of material composition is broadly used in current Hangzhou, it is unclear whether this platform is still applicable to new design projects when some of their materials are replaced with other types. Another issue is the decision-making process on whether the selected WWR makes enough transparent areas. WWR is not the only factor affecting façade aesthetics. Some architects may still feel challenging to assess a design solution based on one singular large window in the middle of a facade. The third limitation is that this study does not deeply discuss different possibilities of ANN models, structures, and hyperparameters. Readers can find more specific discussions in other related studies (García Kerdan & Morillón Gálvez, 2020).

On the other side, based on other limits that can be broken through in the future, there are four main research opportunities. First, the ANN herein is only useful for office design projects located in Hangzhou, not applicable to other regions. Therefore, this methodology will be repeated in more climate regions to serve more users. Related studies can be found in another research paper (Yong et al., 2019).

Second, to accelerate the whole process and avoid too many complexities, this study is carried out without any urban context. Therefore, in future studies, the surrounding neighborhood context will be investigated and integrated to make the building performance simulation more precise.

Third, there is still a usability gap for normal architects to work with Python IDEs. The author plans to move the optimization process together with 3D visualization entirely online. ANN can be deployed on the web front end (Vyas, 2018) by using the Flask, and the optimization process can also be web-based (Tcheremniak & Sigmund, 2001). After doing that, users worldwide can get access to these tools through the World Wide Web.

Fourth, due to this paper’s scope, optimizing the other four variables is not discussed herein. Although this study focuses on optimizing WWRs on four orientations (X5 to X8, referring to Table 1), with a little of modifications to the Python code, this newly built platform actually can be used to optimize the other decision variables (X1 to X4, referring to Table 1).

References
Amasyali, K., & El-gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies. Renewable and Sustainable Energy Reviews, 81(1), 1192–1205. https://doi.org/10.1016/j.rser.2017.04.095.
Amirhosain, S., & Hammad, A. (2019). Developing surrogate ANN for selecting near-optimal building energy renovation methods considering energy consumption, LCC and LCA. Journal of Building Engineering, 25, 100790. https://doi.org/10.1016/j.jobe.2019.100790.
Asadi, E., daSilva, M. G., Antunes, C. H., Dias, L., & Glicksman, L. (2014). Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application. Energy and Buildings, 81, 444–456. https://doi.org/10.1016/j.enbuild.2014.06.009.
Ascione, F., Ascione, F., Bianco, N., De Stasio, C., Mauro, G. M., & Vanoli, G. P. (2017). Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach. Energy, 118, 999–1017. https://doi.org/10.1016/j.energy.2016.10.126.
Algorithmically optimize window wall ratios by using ANN

Tcherniak, D., & Sigmund, O. (2001). A web-based topology optimization program. Structural and Multidisciplinary Optimization, 22(3), 179–187. https://doi.org/10.1007/s001580100135.

Troup, L., Phillips, R., Eckelman, M. J., & Fannon, D. (2019). Effect of window-to-wall ratio on measured energy consumption in US Office Buildings. Energy & Buildings, 203, 109434. https://doi.org/10.1016/j.enbuild.2019.109434.

Vanhoutteghem, L., Vanhoutteghem, L., Skarning, G. C. J., Hviid, C. A., & Svendsen, S. (2015). Impact of façade window design on energy, daylighting and thermal comfort in nearly zero-energy houses. Energy and Buildings, 102, 149–156. https://doi.org/10.1016/j.enbuild.2015.05.018.

Vyas, H. (2018). Deploy a machine learning model using flask, https://towardsdatascience.com/deploy-a-machine-learning-model-using-flask-da580f8460c (Accessed 4 February 2021).

Wagner, A., Bonduel, M., Pauwels, P., & Rüppel, U. (2020). Representing construction-related geometry in a semantic web context: A review of approaches. Automation in Construction, 115, 103130. https://doi.org/10.1016/j.autcon.2020.103130.

Wikipedia. (2018) Grasshopper 3D. Available at: https://en.wikipedia.org/wiki/Grasshopper_3D (Accessed 17 January 2019).

Xu, X., Feng, G., Chi, D., Liu, M., & Dou, B. (2018). Optimization of performance parameter design and energy use prediction for nearly zero energy buildings. Energies, 11(12), 3252. https://doi.org/10.3390/en11123252.

Yong, S. G., Kim, J.-H., Cho, J., & Koo, J. (2019). Meta-models for building energy loads at an arbitrary location. Journal of Building Engineering, 25, 100823. https://doi.org/10.1016/j.jobe.2019.100823.

Zhai, Y., Yi, W., Yanqiu, H., & Xiaojing, M. (2019). A multi-objective optimization methodology for window design considering energy consumption, thermal environment and visual performance. Renewable Energy, 134, 1190–1199. https://doi.org/10.1016/j.renene.2018.09.024.

Zhao, S. (2019). Using algorithms to designate pre-fabricated wall materials: A case study with two implementation methods. Case Studies in Construction Materials, 10(2018), e00220. https://doi.org/10.1016/j.cscm.2019.e00220.

Zhao, S. (2020). Searching for office buildings’ fenestration geometries with a Bi-phase optimization framework. Science and Technology for the Built Environment, 20(9), 1–22. https://doi.org/10.1080/23744731.2020.1777048.

Appendices
The link to download Appendices 1 and 2 is https://drive.google.com/file/d/1NtUSR-D-OuG0iSnidKi2AiXppk0rvoxWw/view