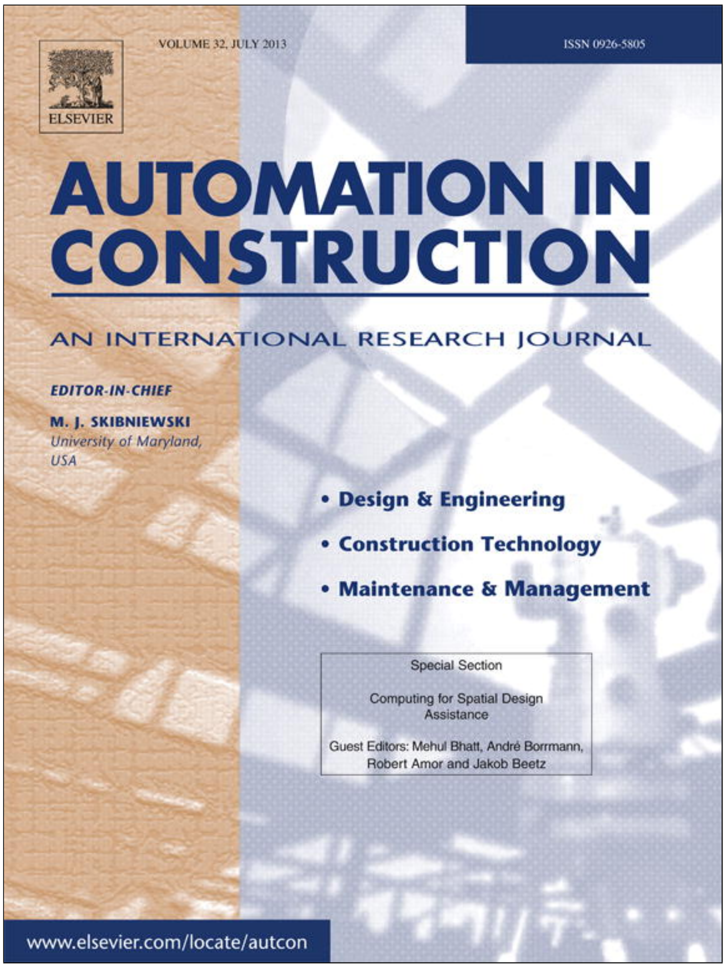

VOLUME 32, JULY 2013
ISSN 0926-5805
ELSEVIER
AUTOMATION IN CONSTRUCTION
AN INTERNATIONAL RESEARCH JOURNAL
EDITOR-IN-CHIEF
M. J. SKIBNIEWSKI
University of Maryland, USA
• Design & Engineering
• Construction Technology
• Maintenance & Management
Special Section
Computing for Spatial Design Assistance
Guest Editors: Mehul Bhatt, André Borrmann, Robert Amor and Jakob Beetz
www.elsevier.com/locate/autcon

Automation in Construction 32 (2013) 125–135

Contents lists available at SciVerse ScienceDirect

Automation in Construction

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

Performance-driven architectural design and optimization technique from a perspective of architects

Xing Shi [a,*], Wenjie Yang [b]

[a] Key Laboratory of Urban and Architectural Heritage Conservation, Ministry of Education, School of Architecture, Southeast University, #2 Si Pai Lou, Nanjing, 210096, China
[b] Tianhua Architecture Design Company Limited, 27 Floor, Zhaofeng Global Mansion, 1800 Zhongshan West Road, Shanghai, 200235, China

ARTICLE INFO

Article history:
Accepted 25 January 2013
Available online 28 February 2013

Keywords:
Performance
Optimization
Simulation
Rhinoceros
Ecotect
Radiance
EnergyPlus

ABSTRACT

Performance-driven architectural design emphasizes on integrated and comprehensive optimization of various quantifiable performances of buildings. As the leading profession of a project team, architects play a vital role in guiding and conducting the performance-driven design. Methodology and techniques start emerging both in literature and practice. However, architects often find them difficult to use for various reasons. Therefore, developing an effective technique to conduct performance-driven design and optimization from the perspective of architects is necessary. This paper starts from discussing the concept of performance-driven architectural design. Existing methodology and techniques are reviewed. The focus is on selecting a basic platform suitable for architects, upon which the technique can be developed. Rhinoceros, an architectural modeling program, is used, along with its graphical algorithm editor Grasshopper, to establish such technique by incorporating three performance simulation programs, namely Ecotect, Radiance, and EnergyPlus. Design cases are presented to demonstrate the technique and its effectiveness.

1. Introduction

Performance-driven architectural design emphasizes on integrated and comprehensive optimization of various quantifiable performances of buildings. It is an important research subject and a design philosophy being practiced by many architects and design firms [1]. Compared with the conventional architectural design methodology, which focuses on space and form, performance-driven design takes a holistic view towards ecological and environmental performances of buildings while ensuring that the functions and esthetics of the design are not overlooked. It is particularly important in countries undergoing fast urbanization such as China [2] since the performance of many newly built buildings affects the overall quality of urbanization.

A major boost for performance-driven architectural design is the implementation of green building standards internationally such as LEED in the US [3], BREEAM in the UK [4], and the newly launched green building evaluation standard in China [5]. These green building standards establish many quantifiable performance requirements to guide and control the design. Thus, performance-driven design is encouraged and more rational thinking and scientific analysis are brought into the field of architectural design. As more and more green buildings emerge, architects, the leading professional of a building project team, urgently need to study and grasp the new design philosophy and the supporting technique to ensure the design quality while keeping the good essence of the conventional design.

1.1. Green buildings and architects

Reviewing the history of green buildings shows a clear pattern that it is largely the research, development, and utilization of new materials and/or mechanical equipments that lead the progress of the field. As a consequence, the green building is gradually becoming a high-tech architectural machine and architects, the supposedly leading professional, are somewhat lost. The conventional architectural design methodology is often powerless facing the scientifically rigorous and quantifiable performance criteria. The implementation of green building standards exacerbates this problem. Two kinds of so-called green architects can be identified.

- The first kind is an architect who follows the conventional design approach to complete the conceptual design. He then turns the work to other professionals such as consulting engineers or mechanical engineers to apply various green technologies. In this way, the green building is designed by rigidly adding technologies without adequate integration and optimization.
- The second kind is an architect who aims at designing a green building in the conceptual stage. Due to the lack of specialized knowledge and technique, he uses conceptual, non-quantifiable,

* Corresponding author. Tel.: +86 25 83790037; fax: +86 25 83790682.
E-mail address: shixing_seu@163.com (X. Shi).

0926-5805/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.autcon.2013.01.015

and sometimes even vague methods to design. The end result is a quite subjective design which may or may not be truly green.

The first kind of green architects already loses control of the design. The final quality and the performance achieved are determined primarily by the effectiveness of technology summation. The cost is significantly higher and more importantly, no integration or optimization of technologies is conducted. The second kind of green architects is still the leading professional in the design team. However, since the design methods he applies are conceptual and non-quantifiable, he has no real control over building performances. His interpretation of the design can only circle around ideas and/or concepts. Performance criteria cannot be rigorously checked, not to mention satisfy green building standards.

Therefore, what architects urgently need is a new methodology and the supporting technique of performance-driven design which can be applied in the early conceptual design stage and are reasonably precise, efficient, and dependable.

1.2. Conventional methodology and performance-driven architectural design

The conventional architectural design methodology is, in essence, an approach involving some basic design principles, mainly based on functions and forms. The driving force is the combination of the architect's rationality and sensibility. When performance criteria must be met, this design methodology is facing unprecedented challenges. Architects have to deal with the following three problems.

- The prerequisite for performance analysis is a building model that can be analyzed. However, the complexity and variance of buildings make an analyzable model quite difficult to obtain. The current practice usually involves setting up a model using design software and then importing the model into performance simulation programs. This process is time-consuming and labor-intensive.
- The model created in most modeling programs only contains geometric information (the latest development and application of building information modeling might change it). Many non-geometric parameters have to be input in the simulation program. This, combined with the previous point, discourages the engineer to use the architect's model for performance simulation purposes. Rather, he prefers to directly set up the model in the simulation program for he can input both geometric and non-geometric information at once.
- However, the modeling capability of most simulation programs is not on the level of commonly used architectural modeling programs, especially when dealing with complex shapes and forms. Here a dilemma arises, i.e., the engineer does not want to use the architect's model because he has to import it and add many parameters before a simulation can be run; on the other hand, the architect is not satisfied with the engineer's simplified model and believes that it lacks details and is not esthetically pleasing.

These three problems are difficult to overcome using conventional design methodology. New approaches and techniques must be developed to assist the architect to carry out performance-driven design.

An architectural design process can be divided into three steps, namely, conceptual design, detailed design, and construction document design. It is widely agreed that design decisions made in the conceptual stage have the largest impact on the final overall performance of the building. Guillemin and Morel conducted a survey on 67 buildings and found that 57% of technological decisions were made in the conceptual design stage, compared with only 13% in the detailed design stage [6]. Therefore, the methodology adopted by the aforementioned two types of green architects clearly has limitations. The right paradigm is to incorporate performance analysis into the early conceptual design stage so that right technical decisions can be made. The performative outcome of different designs should be quantifiable and visible to the client and the architect.

1.3. Performance-driven design and digital technology

Conducting performance analysis and optimizing the design effectively and efficiently used to be challenging. Lately, the rapid progress of digital technology and its application in architecture have changed the field dramatically. The emergence and development of performance simulation tools make rapid performance analysis possible. More and more architects and engineers become familiar with these tools and proactively use them in the design. More powerful personal computers shorten the time needed for analysis. In recent years, building performance simulation has become a very active research field. On one hand, researchers and specialists are studying and producing more powerful simulation tools. On the other hand, practitioners start realizing the value of incorporating them into the design process. This combined force pushes performance-driven architectural design to the forefront.

This paper aims to analyze the concept of performance-driven architectural design and its current status. We argue that the performance-driven architectural design is a design philosophy that must be supported by effective and efficient design technique. Equally important, the technique should be viewed familiar and practical from the perspective of architects. First and foremost, the technique must involve a modeling platform that architects feel comfortable with. Secondly, the simulation tools that can be integrated into the design flow should cover the most important performances that architects need to consider. Last but not least, the optimization algorithms should be readily available. Very few architects can and want to write their own codes for optimization. Such performance-driven design flow and technique can assist the architect to explore many design possibilities and their corresponding performances in a convenient way. The end result is a design that is esthetically pleasing, spatially efficient, and performatively sound.

2. Literature review

Many architects, engineers, and researchers have aimed at achieving performance-driven design. Research works conducted by different professionals show distinguishable features. Analyzing these features helps us understand what kind of approach and technique architects need. The following section presents a short literature review on performance-driven architectural design. The literature is grouped into three categories based on who the leading professional is.

The first category of the research work is led by computer scientists or software engineers. Performance-driven design is achieved by developing source codes from the very bottom. Very few architects are involved since they don't possess the special knowledge and skills to write a significant amount of computer codes. For instance, Ellis et al. [7] developed an automated multivariate optimization tool to perform energy efficient building design. The tool employs multiple modules, including a graphical user interface, a database, a preprocessor, a simulation engine, an optimization engine, and a simulation run manager. All modules are customer written.

The second category of the research work is led by consulting professionals. The framework is to use commercially available optimization program and integrate building performance simulation tools to conduct performance-driven design. Some computer code writing is usually needed to set up the design flow. Shi [8] used modeFRONTIER, a commercially available optimization program, and integrated energy simulation program EnergyPlus to study the optimal insulation strategy for an L-shape, one story building. Also using modeFRONTIER, Manzan and Pinto [9] integrated ESP-r, an energy simulation program, and Radiance, a lighting simulation program, to design an external shading device in an office with a window and different glazing

Table 1
Comparison of the three categories of research work on performance-driven architectural design.

	Category 1	Category 2	Category 3
Leading professional	Computer scientists and software engineers	Building engineers and consultants	Architects
Advantages	Open source code and fast processing	Integrated platform, complete performance optimization	Powerful modeling capability, visible feedback of performance simulation results, familiar to architects
Shortcomings	Difficult coding	User interface not friendly, geometric modeling capability limited	Multiple programs required, slow speed, coding platform limited
Tools used	Coding languages	Commercial optimization programs such as modeFRONTIER and ModelCenter, performance simulation programs such as Ecotect	Geometric modeling programs such as Rhinoceros, performance simulation programs such as Ecotect

characteristics. The same approach can also be found in urban studies. Bruno et al. [10] used modeFRONTIER and integrated CATIA, a modeling program originated from aerospace engineering, to explore a non-linear design space whereby multiple objectives may be optimized concurrently. The authors defined and applied quantitative metrics in order to examine the potential for a new workflow in urban design. Flager and Haymaker [11] compared so-called Multidisciplinary Design Analysis (MDA) and Optimization (MDO) processes in the building and aerospace industries based upon case data gathered on recent projects in each industry. Further, Flager et al. [12] introduced multidisciplinary design and optimization technology and applied it on a classroom building design using another commercially available optimization program called ModelCenter. Chronis et al. [13] integrated climatic and site data into a dynamic model of a large student housing complex project using parametric and optimizing technique.

The above two categories of research work share one commonality, i.e., the leading professionals are not architects. The computer tools involved in the design flow, especially the space and form modeling programs, are either not familiar to architects or capable of generating complex geometric shapes. Therefore, some architects and researchers with an architectural background start exploring performance-driven design and its technology from the perspective of architects. Sargent et al. [14] from the Harvard Design School presented a new approach called SHADERADE. The authors actually developed an eponymous tool to assess the performance of different shading designs. Stravoravdis and Marsh [15] proposed a method to use LUA-scripting to control and manipulate the model and data within a building performance simulation program. A similar approach was taken by Kawakita to optimize the window design [16]. Mark [17] used Bentley's Generative Component and linked it with Ecotect to evaluate whether larger transformations to the structure as a whole or smaller movement in the fabric would help to optimize the solar insolation benefits. The author acknowledged that the technique was initially developed by DeBiswas at MIT. Caldas and Norfork [18] applied genetic algorithm as a generative and search procedure to look for optimized design solutions in terms of thermal and lighting performance in a building. The building performance simulation program used is DOE.

The three categories of research work on performance-driven architectural design share some commonalities and also differ in some ways. Table 1 presents a comparison.

It should be noted that there are currently some ambitious and large-scale research projects in progress that are intended to establish a comprehensive and powerful integrated performance-driven architectural design platform. A noteworthy example is the Virtual Design Studio project funded by the Department of Energy of the US and conducted at Syracuse University [19,20].

Based on the literature review, it is clear that performance-driven architectural design has attracted much attention from architects, engineers, and researchers. As the technology progresses, different approaches have been developed. A new trend is to provide more powerful architectural modeling capability and more user friendly interfaces in the design flow. We believe it is valuable to develop a technique that architects feel comfortable with since they are the leading professional of a design team. In this paper, Rhinoceros, an architectural modeling software, and its script coding component Grasshopper are selected as the platform, based on which a design workflow linking three performance simulation programs, namely Ecotect, Radiance, and EnergyPlus is established.

3. Methodology

3.1. Selection of the design platform

Sketchup, 3dsMax, Maya, etc. are the commonly used modeling programs among architects. Lately, Rhinoceros has gained popularity because of its powerful modeling capability, especially for complex geometric shapes, flexible expandability, and relatively low requirements on computer hardware. In addition, Rhinoceros provides an effective user development platform called Grasshopper, thus enabling architects to customize for some complex projects with special needs.

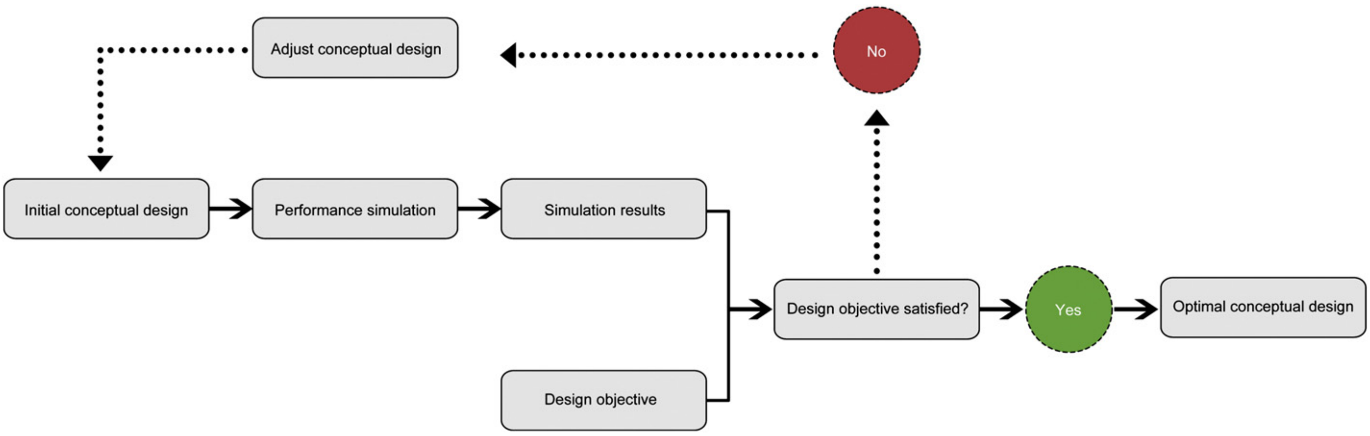


Fig. 1. The workflow of performance driven conceptual design.

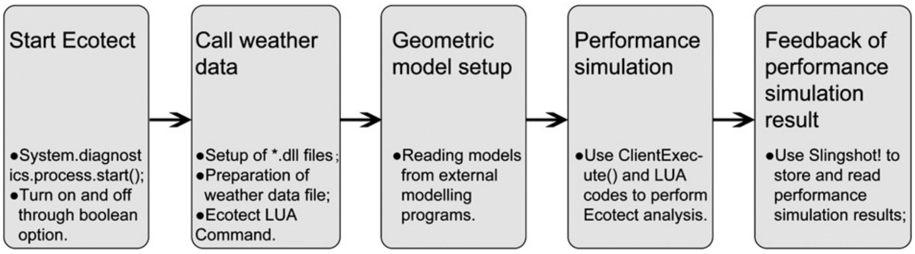


Fig. 2. Diagram of the workflow for integrating Ecotect into Rhinoceros/Grasshopper.

Studying the technique of performance-driven architectural design based on Rhinoceros/Grasshopper is valuable for the following reasons.

- Performance-driven architectural design, while emphasizing on performance optimization, must simultaneously consider space and shape, two of the major design considerations that architects will never neglect. Therefore, the design workflow should incorporate modeling programs familiar to architects. Rhinoceros/Grasshopper is such a program. Design workflow and technique based on it would be friendly to architects.
- The powerful modeling capability of Rhinoceros/Grasshopper makes it an adaptable platform for performance-driven design since it can handle various conceptual designs from linear to non-linear and from simple to complex.

For these reasons, we select Rhinoceros/Grasshopper as the modeling program to study the performance-driven design technique and use Galapagos, a third-party program built in Rhinoceros, to achieve the control of optimization.

3.2. Design workflow

The essence of the Rhinoceros/Grasshopper-based performance driven design technique is to establish a workflow so that performance simulation results can be automatically fed back to the modeling program to guide the design optimization controlled by certain algorithms. Thus, the key to the workflow is a data exchange and communication system to control the entire design and analysis process. Its procedure can be decomposed as follows.

(1) Use the modeling program to generate one or multiple initial designs and parameterize them.
(2) Select the needed geometric, material, physical, and environmental parameters and transport them to the performance simulation program.
(3) Analyze the performance in the simulation program based on the received data.
(4) Compare the simulation results to the predefined optimization objectives. If the objectives were met, end the design workflow. Otherwise, enter the next step.
(5) Automatically, usually driven by an optimization algorithm, or manually adjust the design and repeat the process.

Fig. 1 shows a diagram of the design workflow. It is worth noting that,

- This design workflow is iterative. The number of iterations needed to reach an optimal or desired design depends on various factors, including the initial conceptual design, the optimization objective, the optimization algorithm, and the manual adjustment, etc.

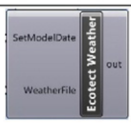
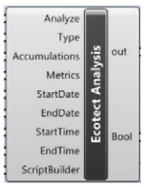
NAME	RUN ECO	WEATHERDATA	MODEL MATERIAL	CALCULATION	RESULT
User-defined Components					
Functions	Use IPC to enable the communication between Rhino /Grasshopper and Ecotect. Start Exotect through the Boolean command.	Define the file path of the weather data and send them to Ecotect by LUA coding.	Decompose the Rhino model and read relevant geometric information. Turn the needed information into a *.scr file based on LUA and call Ecotect to run the *.scr file through IPC process	Set up necessary parameters in Rhino/Grasshopper and generate another *.scr file. Call Ecotect to run the *.scr file to conduct performance simulation through IPC	Run the Boolean command to access the simulation results

Fig. 3. The names and functions of the five user-defined components to integrate Ecotect into Rhinoceros/Grasshopper.

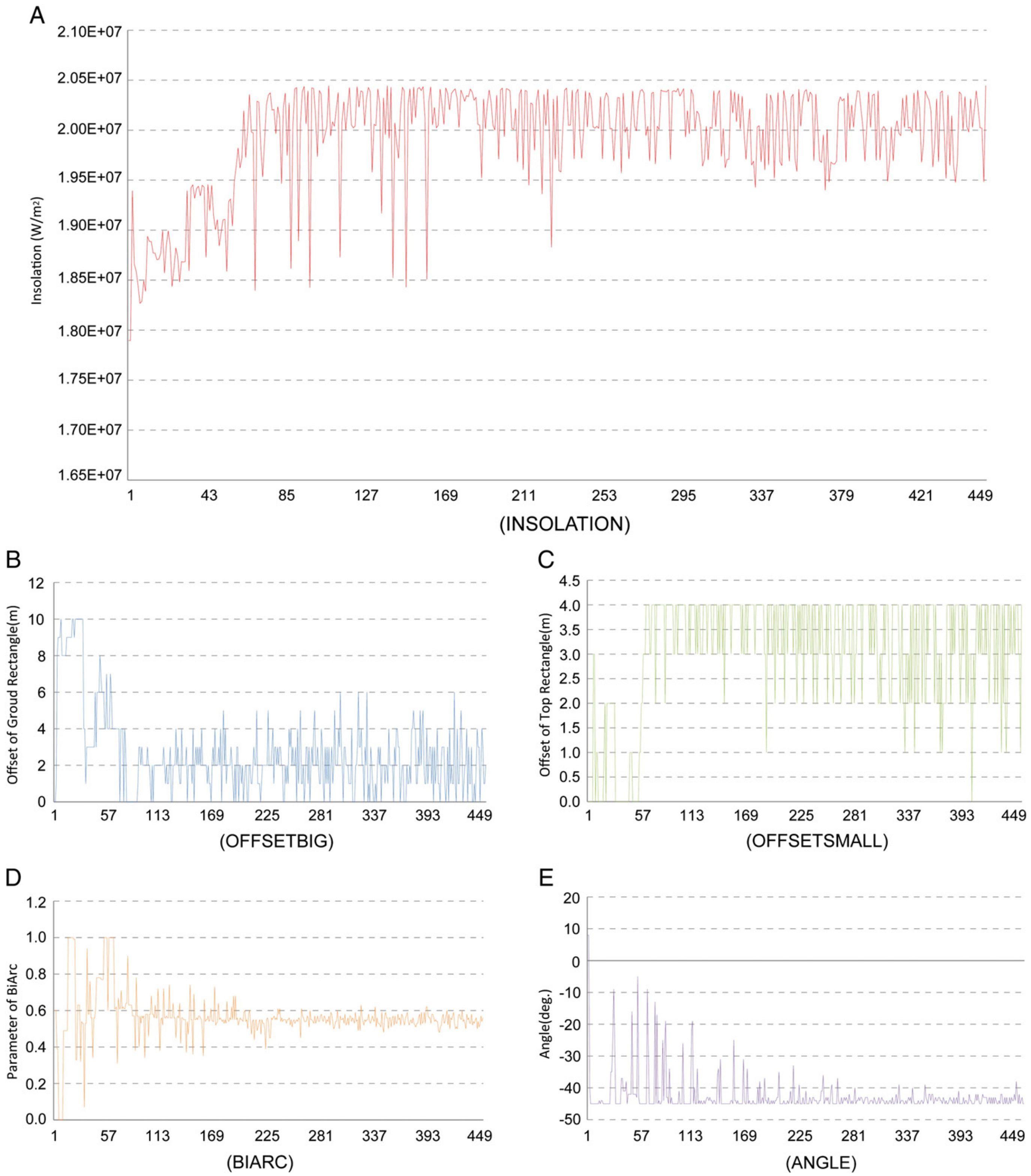


Fig. 4. Total yearly insolation and four parameters defining the roof surface shape.

- The effectiveness of manual adjustment is highly dependent on the operator's knowledge and skills in the field of building performance.
- Automatic adjustment is usually driven by a predefined optimization algorithm.
- The time and resources needed to complete a successful workflow are highly dependent on the complexity of the performance simulation. If the optimized performance is insolation or such that does

Table 2
Design parameters of the 108th conceptual design.

ID	OffsetBig	OffsetSmall	BiArc	Angle	Insolation
108	2	4	0.56	−45	20,445,000

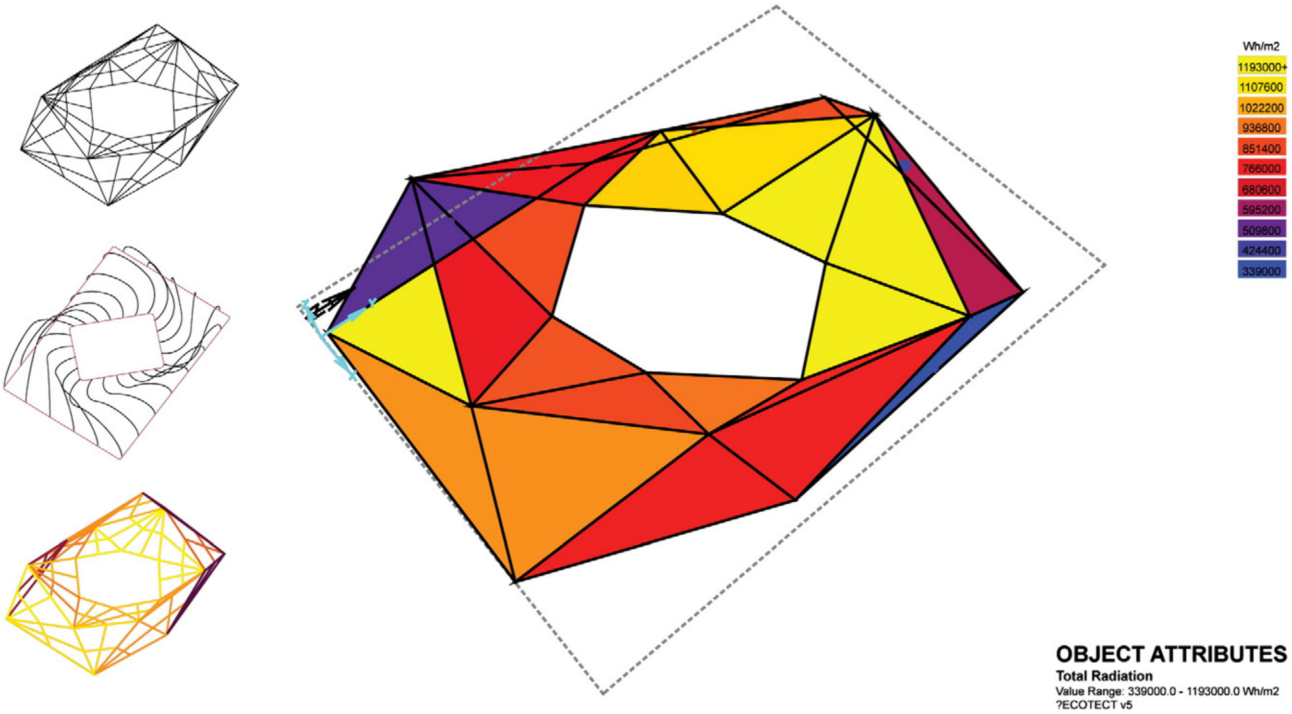


Fig. 5. The final shape of the roof surface with the optimized solar radiation.

not require a complex physical model, the workflow can be straightforward. However, if the architect wants to optimize for some performances that require complex analysis or involves advanced tools such as CFD (Computational Fluid Dynamics) program, the design workflow can be time-consuming and resource intensive. In addition, the pre-defined optimization objectives also affect the time and resources needed to complete a successful workflow. For example, a weak threshold can be quickly reached while a stringent one may take much longer.

The aforementioned design process is achieved by connecting various modules into one coherent workflow. These modules are as follows:

- Graphic User Interface (GUI) module. The GUI module is the beginning of the workflow and the one that architects are most familiar with. It performs functions like modeling geometry, setting up the workflow, inputting commands, graphically showing the performance simulation results, etc. The commonly used GUI modules in architectural design are Rhinoceros, Maya, 3dsMax, Processing, Houdini, etc.
- Optimization module. The optimization module generates new sets of design parameters, usually driven by optimization algorithms, to define a new design and attempt to reach the predefined design objectives.
- Conceptual design generation and data input module. The conceptual design generation and data input module provide capability of generating designs and inputting data. Design generation starts from setting up geometric models, such as mesh, surface, solid, etc. Since performance simulation programs have specific requirements on the format of geometric models, one should select the type of geometric models appropriately. Input data include geometric data, physical data, and algorithm data. Geometric data describe the geometric model and need to be readily transferred into the performance simulation program. Physical data include material properties, environmental parameters, and others that are necessary to conduct performance simulation in addition to geometric data. The data format can be integer, double, long, or string. Algorithm data are important to the optimization process control and the workflow efficiency.
- Communication module. The communication module links the GUI module to the performance simulation module. For close source programs, one needs to establish a client program to enable the communication between processes. In this paper, we call for *Ndde.dll* to achieve the data exchange between Rhinoceros and Ecotect and Radiance. For open source programs such as EnergyPlus, we write DOS batch file to achieve that.
- Performance simulation module. The performance simulation module calculates and analyzes various performances that will be used to

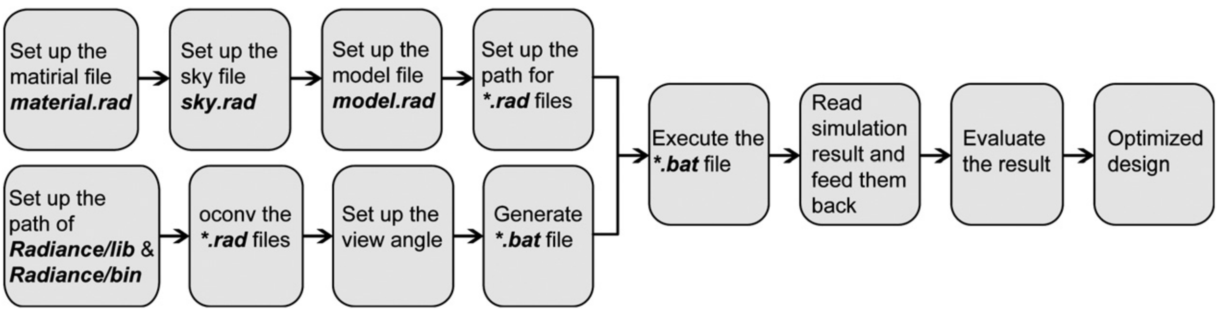


Fig. 6. Workflow of integrating Radiance into Rhinoceros/Grasshopper.

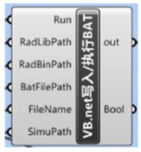
Name	WeatherDATA	MODEL	MATERIAL	POINTS	RIF	BATCH
User -defined Componets						
Functions	Create *sky.rad* file for working environment parameters in Grasshopper using VB.net	Use VB.net to extract the relevant information from Rhino through Grasshopper and create different part of *model.rad* files and assign materials	Use VB.net to create *material.rad* material file in Grasshopper	Use VB.net to extract user-defined point matrix information and create **.pts* file for the reference plane in Grasshopper	Use VB.net to create **.rif* file to control the key parameters that Radiance simulation requires (such as view point, orientation, etc.) in Grasshopper	Use VB.net to create **.bat* file in Grasshopper. Enable the communication between Grasshopper and Radiance through setting the working path and use Boolean to control the execution of simulation

Fig. 7. The names and functions of the six user-defined components to integrate Radiance into Rhinoceros/Grasshopper.

optimize the conceptual design. Most of performance simulations require specialized and complex programs. Commonly used performance simulation programs include energy simulation programs such as EnergyPlus, lighting simulation programs such as Radiance, integrated performance simulation programs such as Ecotect, etc. Most of the performance simulation programs are based on Windows operating system and are closed source. Few are based on Linux or DOS and are open source. For the latter, users can write commands to run applications and obtain more control and flexibility.

- Performance simulation results feedback and analysis module. This module is to store the performance simulation results and feed them back to the optimization algorithm module. For open source programs, the user needs to access the output file and select data needed. When a large number of data are produced, special database may be needed for storage. For instance, the technique presented in this paper involves using Slingshot, an add-in in Grasshopper, to store all relevant data in the workflow.

4. Technique and application of the performance-driven design workflow

The previous section presents a workflow and its modules for performance-drive design. This section discusses in detail the technique of applying such workflow in different situations, namely,

- Integrate Ecotect into Rhinoceros/Grasshopper and study the conceptual design of a roof with a complex geometric shape to achieve maximum insolation,
- Integrate Radiance into Rhinoceros/Grasshopper and study a window design strategy to achieve optimal natural lighting,
- Integrate EnergyPlus into Rhinoceros/Grasshopper and study the conceptual design of a simple building to minimize its energy consumption.

The focus is on establishing the workflow and the technique and demonstrating their applications. In addition, the examples presented, although simplified, all have practical design backgrounds.

4.1. Rhinoceros/Grasshopper and Ecotect integration

The design problem considered originates from a project located in Nanjing, China. It is a building with a complex roof. The geometric shape of the roof is defined using a NURBS surface. For more information on NURBS surfaces and curves, one can refer to [21]. The client wants to integrate a photovoltaic system on the roof and produce electricity as much as possible. Thus, the shape of the roof surface becomes a critical design issue and is obviously performance-driven. The problem would be much easier if the roof had a regular shape. With a complex shape like NURBS surface, it is quite challenging. A technique of integrating Ecotect into Rhinoceros/Grasshopper was developed and a performance-driven workflow was established to assist the architect to determine the optimal roof surface shape.

To integrate Ecotect into Rhinoceros/Grasshopper and create the workflow, we used Inter-process Communication in Windows system to enable communication between processes and simultaneously sent commands from Rhinoceros/Grasshopper to Ecotect. In this way,

Table 3
Material reflectance and optimization parameters.

Material reflectance (%)				Optimization parameters				
Ground	Ceiling	Interior wall	Window (transparency in %)	Max. stagnant	Population	Initial boost	Maintain	Inbreeding
20	90	50	70	20	10	1	5%	75%

Ecotect was kept running in the background and the modeling environment, i.e. the user interface of Rhinoceros, that the architect is familiar with was kept in the foreground. This is an important advantage that should not be overlooked. The workflow is shown in Fig. 2. Note that LUA in Fig. 2 is a scripting language used with Ecotect.

The workflow was run in Grasshopper via user-defined components written in VB.net. There are five user-defined components, namely RUN ECO, WEATHERDATA, MODEL & MATERIAL, CALCULATION (OBJ or GRID), and RESULT. Their names and functions are summarized in Fig. 3. Note that IPC in Fig. 3 means Inter-Process Communication.

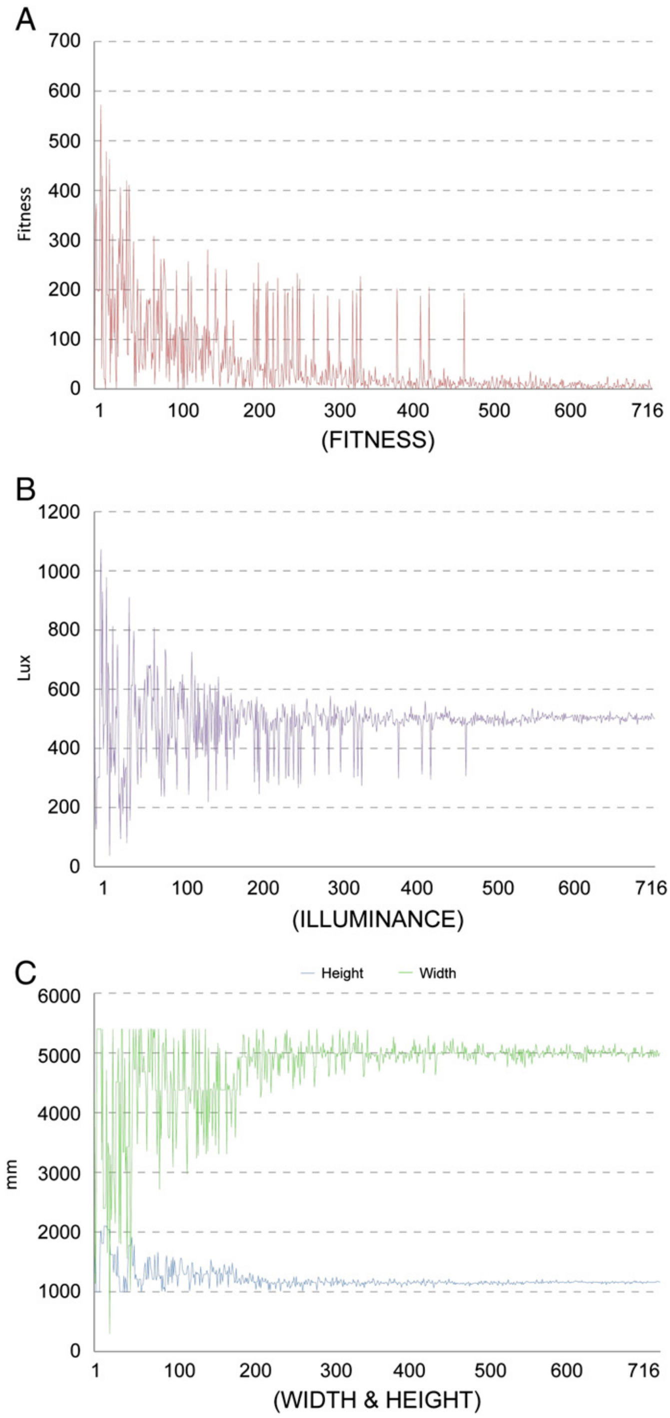


Fig. 8. Fitness, illuminance, and width and height of the window.

The workflow uses four parameters, namely OffsetBig, OffsetSmall, BiArc, and Angle to control the roof surface shape. The driving performance criterion is to obtain the largest total yearly insolation on the roof which is simulated by Ecotect. After initiating the workflow, the user can observe the converging curve and decide when to terminate the design process or let the program stop at the preset number of iterations. Fig. 4a shows a plot of the total yearly isolation of each generated conceptual design. Fig. 4b, c, d, and e shows the plots of the four geometric parameters defining the roof surface shape, respectively.

Based on the observation of the data, the conceptual design with the largest yearly insolation is the 108th one. Its corresponding design parameters are shown in Table 2. The final shape of the roof surface is shown in Fig. 5.

4.2. Rhinoceros/Grasshopper and Radiance integration

Using natural lighting creatively has long been one of the favorite design approaches for architects since it provides physical lighting and offers spiritual values as well. Natural lighting is also an effective way to minimize energy consumption of buildings. The design problem considered is a simple box shape residential building with a 6 m × 6 m floor plan. The design objective is to find the window design, i.e. height and width, to satisfy the requirement that the average daylight illuminance at 900 mm above the floor is 500 lx, a standard requirement in Chinese code for residential buildings. Note that the Chinese standard requires that the illuminance due to natural lighting shall be analyzed using a fully overcast sky model. Therefore, the orientation of the window, assuming only one window for the building, is not a factor in this case.

The development of natural lighting simulation tools can be dated back to the 1970s. Nowadays, commonly used programs include Lumen Micro 2000, AGI32, Lightscape, Ecotect, Radiance, etc. Radiance was selected to demonstrate the workflow partly because it is open source.

As an advanced daylighting simulation program, Radiance is not a common tool for architects. It does not have a graphic user interface and requires an accurate model with many parameters for simulation purposes. The technique presented below uses Rhinoceros/Grasshopper to set up the model, materials, and other simulation parameters. Grasshopper makes it possible to add more parameters to the workflow as the design and simulation evolve. In other words, one can start from a simplified model and gradually make it more complex and closer to the real model. Fig. 6 shows a diagram of the workflow.

We used VB.net to define modules that can automatically generate parameters in Rhinoceros/Grasshopper. Files generated include **.rad* file, **.rif* file, and **.bat* file. The aim is to enable Radiance to be running in the background while Rhinoceros/Grasshopper is running in the foreground. To generate the required parameters for simulation, six components were created in Grasshopper. Their names and functions are shown in Fig. 7.

The input parameters are height and width of the window. The output parameter is the average illuminance on the plane at 900 mm above the floor. The objective is to find the window design in which the output parameter is no less than 500 lx. Other important parameters are shown in Table 3.

The workflow was initiated and terminated when fitness showed a clear trend of converging. Fig. 8a, b, and c shows the plots of fitness,

Table 4
Design parameters of the 547th and 636th design.

Design ID	Window width (mm)	Window height (mm)	Average illuminance (lux)	Fitness
547/636	4992	1161	500	0.006051

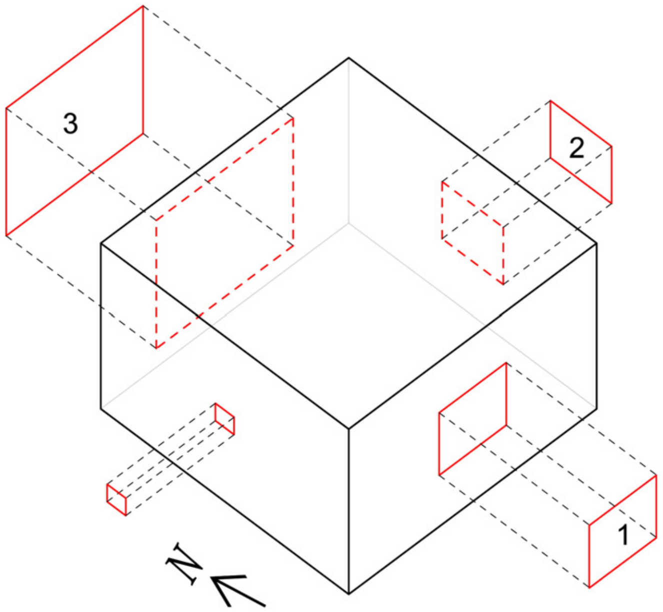


Fig. 9. The rectangular building and its four windows.

average illuminance, window height and width, respectively. Table 4 shows the two optimal designs and their parameters. The window height and width are rounded to be 5000 mm and 1200 mm, respectively.

4.3. Rhinoceros/Grasshopper and EnergyPlus integration

Energy consumption is one of the primary concerns for green building design. EnergyPlus is a standard energy simulation engine developed by the US Department of Energy. EnergyPlus is open source and provides detailed descriptions, engineering references, examples, etc. in a freely downloaded package. This section presents a technique of integrating EnergyPlus into Rhinoceros/Grasshopper and setting up a performance-driven workflow with similar features as discussed previously.

The considered design problem is a rectangular building with three dimensions of 4200×4200×3000 mm, located in Nanjing, China. The building has one window on each wall with a constant total window area. Fig. 9 illustrates the building and its four windows. The objective is to find the window area on each wall to minimize the energy consumption. Let A_S, A_E, A_W, A_N denote the window area on the south, east, west, and north walls respectively. The total window area on four walls is set to be 7.29 m^2. The following equations are used in the workflow to determine each window area:

$$\begin{aligned} A_S &= s\cdot 7.29 \\ A_E &= e\cdot(7.29-A_S) \\ A_W &= w\cdot(7.29-A_S-A_E), \\ A_N &= 7.29-A_S-A_E-A_W \end{aligned} \quad (1)$$

where *s*, *e*, *w* are numbers between 0.01 and 0.99.

The technique of integrating EnergyPlus into Rhinoceros/Grasshopper is similar to that presented previously in the Radiance section. VB.net was used to define components and automatically generate parameters. The EnergyPlus input file, i.e. **.idf* file, contains a lot of information. To demonstrate the capability of the technique, we only created four components, namely ZONE, GEOMETRY, RUN PERIOD, and PROJECT INFORMATION. Fig. 10 summarizes the four components and their functions.

The optimization results are shown in Fig. 11. Fig. 11a is the energy consumption by assuming an ideal load system with a COP of 1. Fig. 11b, c, d, e shows the window areas of the south, east, west, and north walls, respectively. It is clear, based on Fig. 11a, that the overall energy consumption converges to a minimum value during the optimization process. One can select the best window design strategy based on the optimization results and other practical design considerations. In this case, the solution at Step 750, shown in Fig. 11a, has the minimum overall energy consumption. Its corresponding window design is 0.233 m2 for the east window, 0.046 m2 for the south window, 6.16 m2 for the west window, and 1.4 m2 for the north window. At his point, the architect can make a practical decision of not designing any window on the south elevation since the area is approximately zero and designing the other three windows as shown above. The final design may not be strictly optimal but should be satisfactorily close and practical.

5. Conclusions and future work

This paper studies the concept, methodology, and technique of performance-driven architectural design from the perspective of

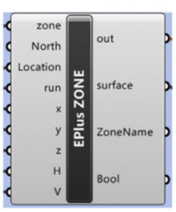
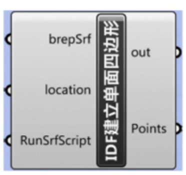
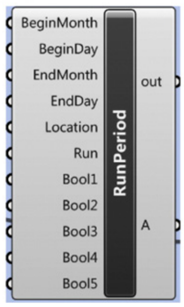
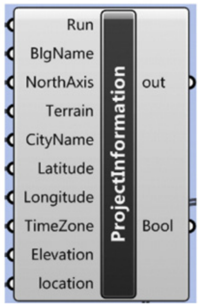
Name	Zone	Geometry	Run Period	Project Information
User -defined Componets				
Functions	Set up zones in accordance with Rhino model and define orientation	Generate geometric model in accordance with Rhino model	Define simulation time	Define project information

Fig. 10. The names and functions of the four user-defined components to integrate EnergyPlus into Rhinoceros/Grasshopper.

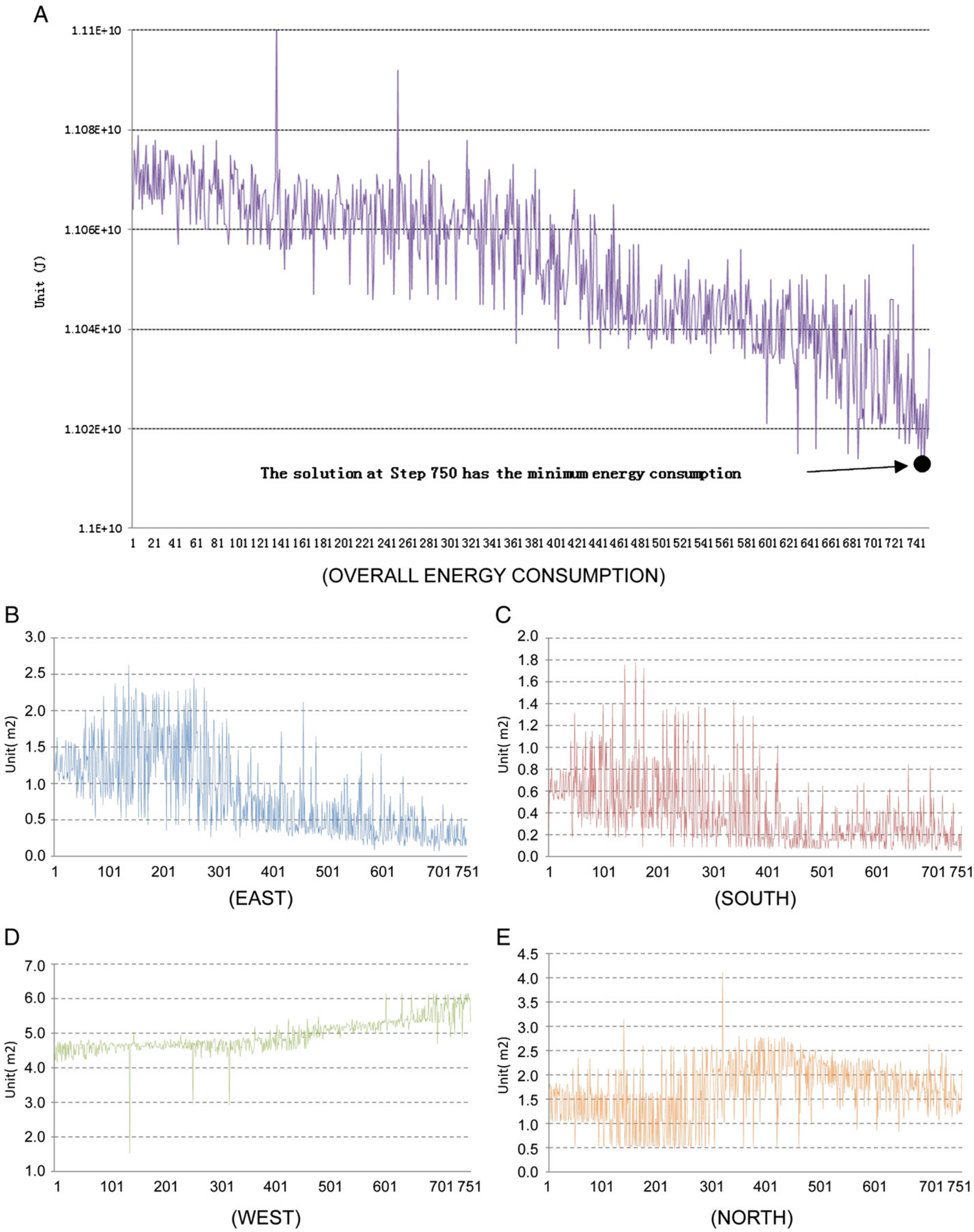


Fig. 11. Overall energy consumption and the window areas on four walls.

architects. The main arguments and conclusions are summarized as follows:

- Performance-driven architectural design is attracting more and more attention from both practicing architects and researchers. It is a response to the movement of green and sustainable design.
- The conventional architectural design methodology cannot be fully performance-driven because it primarily focuses on space, form, functions, and esthetics while performance-driven design needs to develop and refine the design based on scientifically sound performance analysis.
- The methodology and technique for performance-driven design must address the needs of architects since they are the leading professional in a design team. Therefore, Rhinoceros/Grasshopper is considered an appropriate platform upon which such technique and workflow can be established.
- Three important and commonly used performance simulation programs, namely Ecotect, Radiance, and EnergyPlus, were integrated into Rhinoceros/Grasshopper to establish the workflows for performance-driven architectural design and optimization. Some program development and code writing capability were needed.
- Each one of the three techniques and workflows was demonstrated using one practical design case. It is clear that they can facilitate the performance-driven design and assist the architect to make sound design decisions.

Although the methodology and techniques developed in the paper are valuable, they have some limitations that warrant future research work. First, the workflows were set up using customer written computer codes which are not familiar to architects. It should be worthwhile to standardize the codes and develop architect-friendly interface so that no coding capability is required. Secondly, the optimizations demonstrated are all single objective. However, most, if not all, architectural design problems are multi-objective. Therefore, expanding the technique from single objective to multi-objective is a natural next step.

Acknowledgment

The financial support of this paper is provided by the National Natural Science Foundation of China (51008058) and Ph.D. Programs Foundation of Ministry of Education of China (20100092120001).

References

[1] X. Shi, Performance-based and performance-driven architectural design and optimization, Frontiers of Architecture and Civil Engineering in China 4 (4) (2010) 512–518.
[2] C. Gu, L. Wu, I. Cook, Progress in research on Chinese urbanization, Frontiers of Architectural Research 1 (2) (2012) 101–149.
[3] United States Green Building Council, LEED — New Construction and Major Renovation V2.2 Reference Guide. Washington D.C. , 2007.
[4] D. Crawley, I. Aho, Building environmental assessment methods: applications and development trend, Building Research & Information 27 (4–5) (1999) 300–308.
[5] Ministry of Construction, China, Evaluation Standard for Green Building, GB/T 50378. Beijing; 2006. In Chinese, 2006.
[6] A. Guillemin, N. Morel, An innovative lighting controller integrated in a self-adaptive building control system, Energy and Buildings 33 (5) (2001) 477–487.
[7] P. Ellis, T. Griffith, N. Long, P. Torcellini, D. Crawley, Automated multivariate optimization tool for energy analysis, IBPSA SimBuild 2006 Conference, Cambridge, Massachusetts, 2006.
[8] X. Shi, Design optimization of insulation usage and space conditioning load using energy simulation and genetic algorithm, Energy 36 (3) (2011) 1659–1667.
[9] M. Manzan, F. Pinto, Genetic optimization of external shading devices, The 11th International IBPSA Conference, Glasgow, Scotland, 2009.
[10] M. Bruno, K. Henderson, H. Kim, Multi-objective optimization in urban design, Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design, San Diego, USA, 2011.
[11] F. Flager, J. Haymaker, A comparison of multidisciplinary design, analysis and optimization processes in the building construction and aerospace industries, Conference on Information Technology in Construction, 2007, pp. 625–630.
[12] F. Flager, G. Soremekun, B. Welle, J. Haymaker, P. Bansal, Multidisciplinary process integration & design optimization of a classroom building, CIFE Technical Report #TR175, Stanford University, 2008.
[13] A. Chronis, K. Liapi, I. Sibetheros, A parametric approach to the bioclimatic design of large scale projects: the case of a student housing complex, Automation in Construction 22 (2012) 24–35.
[14] J. Sargent, J. Niemasz, C. Reinhart, Shaderade: combining Rhinoceros and EnergyPlus for the design of static exterior shading devices, Building Simulation 2011, Sydney, Australia, 2011.
[15] S. Stravoravdis, A. Marsh, A proposed method for generating, storing and managing large amounts of modeling data using scripts and on-line databases, The 9th International IBPSA Conference, Montreal, Canada, 2005.
[16] G. Kawakita, Environmental Optimisation Methods in Sustainable Design Process: In Combination with Evolution-Based Digital Technology. Master thesis, Oxford Brooks University, 2008.
[17] E. Mark, Optimizing solar insolation in transformable fabric architecture: a parametric search design process, Automation in Construction 22 (2012) 2–11.
[18] L. Caldas, L. Norford, A design optimization tool based on a genetic algorithm, Automation in Construction 11 (2) (2002) 173–184.
[19] P. Pelken, J. Zhang, W. Feng, Y. Chen, D. Rice, L. Gu, H. Henderson, Virtual design studio — part 1: design process, Proceedings of CHAMPS 2011: The 8th International Forum and Workshop on Combined Heat, Air, Moisture and Pollutant Simulations, Nanjing, China, 2011.
[20] J. Zhang, P. Pelken, W. Feng, Y. Chen, D. Rice, L. Gu, H. Henderson, Virtual design studio — part 2: platform design, Proceedings of CHAMPS 2011: The 8th International Forum and Workshop on Combined Heat, Air, Moisture and Pollutant Simulations, Nanjing, China, 2011.
[21] L. Piegl, W. Tiller, The NURBS Book, Springer, Germany, 1996.