Dynamic and interactive re-formulation of multi-objective optimization problems for conceptual architectural design exploration

Yang, Ding; Di Stefano, Danilo; Turrin, Michela; Sariyildiz, Sevil; Sun, Yimin

DOI
10.1016/j.autcon.2020.103251

Publication date
2020

Document Version
Final published version

Published in
Automation in Construction

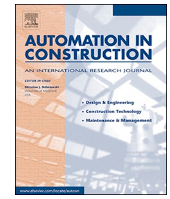
Citation (APA)
Yang, D., Di Stefano, D., Turrin, M., Sariyildiz, S., & Sun, Y. (2020). Dynamic and interactive re-formulation of multi-objective optimization problems for conceptual architectural design exploration. *Automation in Construction*, *118*, Article 103251. https://doi.org/10.1016/j.autcon.2020.103251

Automation in Construction 118 (2020) 103251

Contents lists available at ScienceDirect

Automation in Construction

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



Dynamic and interactive re-formulation of multi-objective optimization problems for conceptual architectural design exploration

Ding Yang[a,b,c], Danilo Di Stefano[d], Michela Turrin[c], Sevil Sariyildiz[c], Yimin Sun[a,b,*]

[a] School of Architecture, South China University of Technology, Guangzhou, China
[b] State Key Laboratory of Subtropical Building Science, South China University of Technology, Guangzhou, China
[c] Chair of Design Informatics, Department of Architectural Engineering and Technology, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, the Netherlands
[d] ESTECO SpA, Trieste, Italy

ARTICLE INFO

Keywords:
Simulation-based multi-objective optimization
Conceptual architectural design
Optimization problem re-formulation
Divergent concept generation
Information and knowledge extraction
Self-organizing map
Hierarchical clustering
Hierarchical variable structure
GH-MF integration
Top-daylighting system

ABSTRACT

Simulation-Based Multi-Objective Optimization (SBMOO) methods are being increasingly used in conceptual architectural design. They mostly focus on the solving, rather than the re-formulation, of a Multi-Objective Optimization (MOO) problem. However, Optimization Problem Re-Formulation (Re-OPF) is necessary for treating ill-defined conceptual architectural design as an iterative exploration process. The paper proposes an innovative SBMOO method which builds in a dynamic and interactive Re-OPF phase. This Re-OPF phase, as the main novelty of the proposed method, aims at achieving a realistic MOO model (i.e., a parametric geometry-simulation model which includes important objectives, constraints, and design variables). The proposed method is applied to the conceptual design of a top-daylighting system, focusing on divergent concept generation. The integration of software tools Grasshopper and modeFRONTIER is adopted to support this application. The main finding from this application is that the proposed method can help to achieve quantitatively better and qualitatively more diverse Pareto solutions.

1. Introduction

1.1. Context

1.1.1. Necessity of requirement and concept re-definition

Conceptual architectural design is the early stage of architectural design, where knowledge is lacking and a number of issues are ill-defined (or ill-structured). This design stage mainly aims at finding a promising design concept that is most likely to meet all important architectural and engineering requirements. Here, a design concept refers to a combination of ideas about the form, technology, working principles of an artefact being designed, namely about how the artefact may satisfy related requirements [1]. A design concept can be explored through vertical or lateral transformation [2,3], thus deriving a vertical or lateral concept accordingly. Vertical transformation focuses on refining existing ideas, while lateral transformation focuses on enriching new ideas. The requirements to be satisfied include quantitative and qualitative requirements [4,5]. In conceptual architectural design, substantial knowledge is required, such as knowledge about various requirements, broad concepts, and their interplay, etc. Unfortunately, in this design stage, such kind of knowledge is often insufficient. Designers may have limited understanding of what should be treated as important requirements, promising concepts, and how the requirements and concepts may affect each other, etc. Their understanding of these issues may become even more limited, when confronted with many conflicting requirements and competing concepts. As a result, initial requirements and concepts are often vague, uncertain, and incomplete; more generally speaking, a number of issues in conceptual architectural design are ill-defined [6].

Treating design, especially the early design stage, as an iterative exploration process, can help to acquire desired knowledge and deal with ill-defined issues. The idea of treating design as exploration has been put forward by some precedent studies [7–10]. The exploration can be defined, among other definitions, as a phenomenon in design

Abbreviations: DoE, design of experiments; GH, McNeel's Grasshopper; HC, Hierarchical clustering; Initial-OPF, Optimization problem initial-formulation; MF, ESTECO's modeFRONTIER; MOO, Multi-objective optimization; OPF, Optimization problem formulation; OPS, Optimization problem solving; Re-OPF, Optimization problem re-formulation; SBMOO, Simulation-based multi-objective optimization; SOM, Self-organizing map

* Corresponding author.
E-mail addresses: yd52503028@gmail.com (D. Yang), distefano@esteco.com (D. Di Stefano), M.Turrin@tudelft.nl (M. Turrin), I.S.Sariyildiz@tudelft.nl (S. Sariyildiz), arymsun@scut.edu.cn (Y. Sun).

https://doi.org/10.1016/j.autcon.2020.103251
Received 28 July 2019; Received in revised form 5 April 2020; Accepted 29 April 2020
Available online 01 July 2020

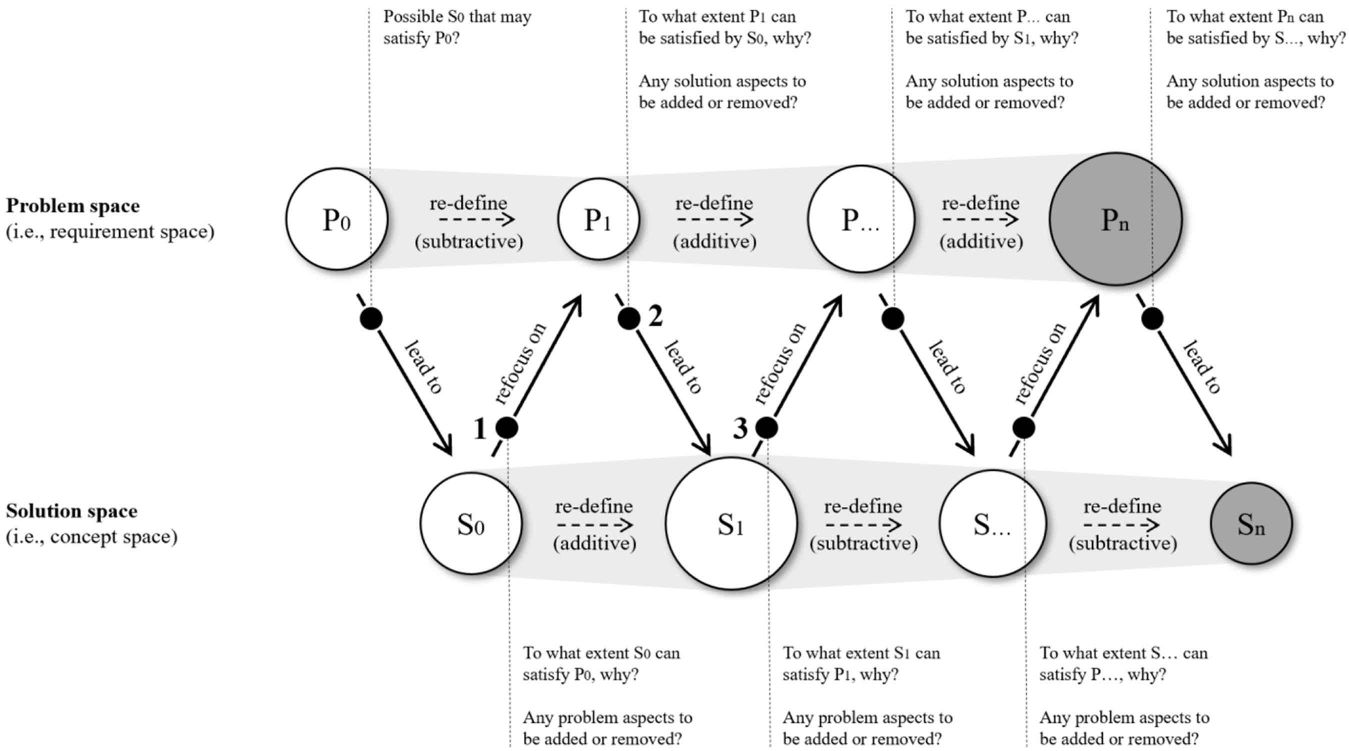


Fig. 1. Co-evolution of problem space and solution space, adapted from [11–13]. The varying sizes of circles show the additive and subtractive re-definition of problem space and solution space; the dots show the knowledge extraction.

where problem space (i.e., requirement space) interacts and evolves with solution space (i.e., concept space) over time [11–13], that is, co-evolution of problem space and solution space (Fig. 1). Specifically, the exploration is an iterative and situated process where designers interpret problems, propose solutions, and more importantly, re-define the problems and/or solutions [14]. The "re-definition" is key to acquiring new knowledge, as shown in Fig. 1 and described below. After the initial definition of problem and solution space (i.e., P_0 and S_0), (1) knowledge that designers possess (i.e., dot 1) can trigger a move to refocus on a re-defined problem space (i.e., P_1); (2) new knowledge (i.e., dot 2) can be shaped through designers' understanding on previous problem and solution space (i.e., P_1 and S_0); (3) knowledge that designers possess (i.e., dot 2) can trigger a further move to lead to a re-defined solution space (i.e., S_1), (4) new knowledge (i.e., dot 3) can be shaped through designers' understanding on previous problem and solution space (i.e., P_1 and S_1); and the remaining actions continue in a similar manner, until knowledge acquired has become insignificant or designers' understanding cannot change enough to warrant further re-definition [8]. As described, the re-definition and knowledge extraction intertwine closely and proceed in an alternating fashion. In this way, designers can acquire and accumulate knowledge, gradually better understand problems and/or solutions in design, and eventually better re-define final problem and solution space (i.e., P_n and S_n). In this sense, the exploration can be also seen as an open-ended human learning process. Moreover, the re-definition can be done in two ways: an additive way (i.e., divergent enrichment), and a subtractive way (i.e., convergent refinement). The two ways of the re-definition can occur throughout all design stages; but, the earlier the design stage, the more meaningful to encourage the divergent enrichment [15], and to broaden the scope of the exploration.

Obviously, it is reasonable to treat the ill-defined conceptual architectural design as the iterative exploration. In such design exploration, it is necessary to allow continuous re-definition of (quantitative and qualitative) requirements and (vertical and lateral) concepts, so as to achieve more precise, certain, and complete final requirements and concepts. Importantly, given the early design stage, divergent enrichment of requirements and concepts is worth to be encouraged, especially revealing lateral concepts.

1.1.2. *Necessity of optimization problem re-formulation (Re-OPF)*

Performativity can be seen as the fourth dimension in architectural design [16]. Performative design [17–20], or performative architecture [21–25], has become a prevailing paradigm in the field of computer-aided conceptual architectural design. In this paradigm, simulation-based optimization [26] is used to achieve optimal designs satisfying architectural and engineering performance requirements. It is known as Simulation-Based Multi-Objective Optimization (SBMOO) when taking advantage of Multi-Objective Optimization (MOO) [27].

Optimization Problem Re-Formulation (Re-OPF) is a necessity to achieve a right or more realistic MOO problem from a design task. The conversion from a design task to an MOO problem and an MOO model, is always a matter of the first priority. Here, an MOO problem refers to a descriptive statement of objectives, constraints, and design variables for a design task; an MOO model refers to a mathematical expression of an MOO problem (i.e., a parametric geometry-simulation model which includes objectives, constraints, and design variables). It is important to know that an MOO problem is in fact an approximation or partial representation of a real design task; thus, there is a gap between the MOO problem and the real design task [28]. This gap is often large in the early design stage, as the initial understanding of the design task is very limited and many issues are still ill-defined; thus, the gap should be reduced as much as possible, to prevent the computation of unrealistic or unfeasible solutions [28]. Given these facts, Optimization Problem Initial-Formulation (Initial-OPF) should be re-formulated and improved as much as possible, in order to achieve a MOO problem which can better approximate the real design task. The necessity of Re-OPF can be also understood as a consequence of re-defining requirements and concepts. That is, once requirements and concepts of a design task are re-defined, associated objectives, constraints, and design variables should be re-formulated accordingly.

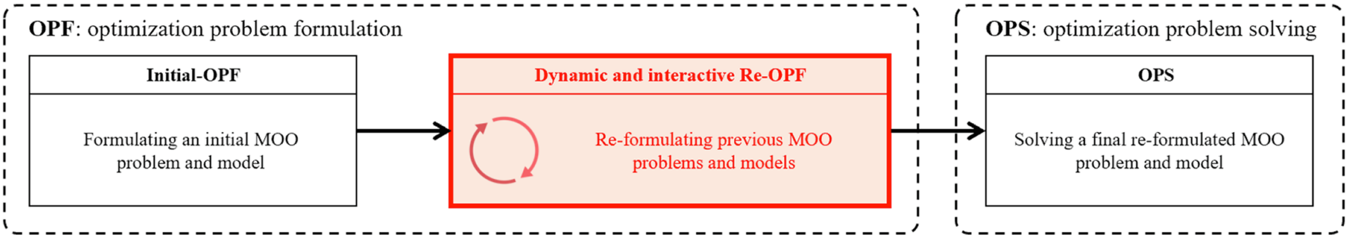


Fig. 2. The proposed SBMOO method. The red box shows the main novelty of the method - the dynamic and interactive Re-OPF phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

SBMOO methods can be used to aid the conceptual architectural design exploration. In such design exploration, it is necessary to allow continuous re-formulation of objectives, constraints, and design variables, so as to achieve a more realistic MOO problem. Specifically, allowing such re-formulation can be broken down into three specific demands: (1) allowing adding and/or removing objectives, constraints, and design variables multiple times; (2) allowing considering objectives and constraints for quantitative and qualitative requirements; (3) allowing considering design variables for vertical and lateral concepts. Importantly, given the early design stage, divergent enrichment of objectives, constraints, and design variables is worth to be encouraged, especially revealing design variables for lateral concepts.

The above demands indicate two required characteristics of Re-OPF. First, Re-OPF should be dynamic (i.e., dynamic Re-OPF). Second, Re-OPF should be interactive (i.e., interactive Re-OPF which leverages both human roles and computer roles). On one hand, human roles are crucial, including human creativity (i.e., a divergent thinking style [29] leading to creativity) and human subjectivity (i.e., Kansei aspects [30] like intuitions, preferences etc.). Human creativity can facilitate divergent enrichment of requirements and concepts; human subjective interpretation of information is required for identifying important requirements and promising concepts [31]. On the other hand, computer roles are complementary to human roles. Computation power can augment, but not replace, human creativity by alleviating routine work [32]; computational objective analysis of data is useful for extracting information about requirements and concepts [33].

1.2. Problem

Although SBMOO methods have been increasingly used to support conceptual architectural design in recent years [34–36], there are still a number of shortcomings. A major one is that, in general, existing SBMOO methods have largely been “cut and paste” from the field of detailed engineering design, with little-to-no adaptation for the needs required in conceptual architectural design [37], as specified below.

First, existing SBMOO methods mostly do not focus on Re-OPF [33]. This can be due to the use of an unrealistic assumption in conceptual architectural design optimization. For the sake of simplicity, ill-defined conceptual design has often been unrealistically assumed to be a well-defined stage where all objectives, constraints, and design variables are given (i.e., not replaceable or removable). Thus, related methods often focus on solving a known MOO problem - Optimization Problem Solving (OPS), rather than finding an unknown MOO problem - Optimization Problem Formulation (OPF).

Second, there are a small number of SBMOO methods which focus on Re-OPF, but most of them do not allow dynamic Re-OPF and interactive Re-OPF simultaneously. The lack of dynamic Re-OPF can be due to the underestimation of the constant nature of Re-OPF. Conceptual design may be seen as an ill-defined stage, but not necessarily an open-ended stage. Thus, related methods only conduct re-formulation once. The lack of interactive Re-OPF can be due to the underestimation of human roles for Re-OPF. Designer subjectivity may be completely excluded, and/or, designer creativity may be constrained within a narrow design direction. Thus, related methods cannot consider objectives for qualitative requirements, and/or, reveal design variables for lateral concepts.

Third, some techniques and tools are useful for supporting SBMOO methods which focus on Re-OPF, but they still face some challenges. These challenges include, for instance, ensuring proper flexibility of parametric models, ensuring proper use of analysis techniques, and testing the usability of the tools, especially for supporting SBMOO methods which involves dynamic Re-OPF.

1.3. Aim

This paper aims to propose an innovative SBMOO method which allows dynamic Re-OPF and interactive Re-OPF simultaneously and is suitable for use in conceptual architectural design exploration. The main novelty of the proposed method is a dynamic and interactive Re-OPF phase built in the OPF phase (Fig. 2). The Re-OPF phase has two characteristics: a dynamic Re-OPF characteristic which can enable MOO problems to be re-formulated multiple times; and, an interactive Re-OPF characteristic which can enable MOO problems to be re-formulated by considering qualitative objectives, and/or, revealing lateral concept related variables. The Re-OPF phase is realized through three groups of looped actions: data generation, information and knowledge extraction, and MOO model re-formulation. The information and knowledge extraction are crucial and connect the other two groups of actions. Namely, data generation is the basis of the information and knowledge extraction; and, MOO model re-formulation is supported by the information and knowledge extraction. In this paper, divergent enrichment of concepts (i.e., divergent concept generation [38]) is emphasized, aiming to achieve quantitatively better and qualitatively more diverse Pareto solutions.

To implement the proposed method, relevant computational techniques and a promising integration of software tools are adopted. The techniques are used to implement the actions needed for the proposed method. They include: frequently highlighted techniques, like geometric parametric modelling [39], multi-disciplinary simulation modelling [40], MOO, which may now be used in a more advanced or different manner; and seldom highlighted techniques, like Design of Experiments (DoE) [41], correlation analysis [42], cluster analysis [43], etc., which are now important for the Re-OPF phase. A promising integrated tool is used to implement the techniques needed for the proposed method. It is called “GH-MF” which integrates McNeel's Rhinoceros with Grasshopper (GH) [44,45] and ESTECO's modeFRONTIER (MF) [46]. It was developed based on a collaboration between TU Delft and ESTECO [33,47].

To verify the capability of the proposed method, and examine the usability of the techniques and tools, a case study of the conceptual design of a top-daylighting system is conducted. The case mainly focuses on divergent enrichment of three typical types of top-daylighting concepts (i.e., skylights, roof monitors, and saw-tooth clerestories [48]) based on daylight, energy and aesthetic performances. By applying the proposed method, each of the initial concepts is enriched to form several modified concepts (i.e., lateral concepts that include new features);

Table 1
Review of SBMOO methods which focus on Re-OPF. Note: "*" marks the classification criteria.

Type	Literature	Application field	Initial quantitative objectives and constraints	Initial qualitative objectives and constraints	Initial design variables	Ways of Re-OPF	Dynamic Re-OPF *	Interactive Re-OPF *
Type 1	Heiselberg et al. [51]	Conceptual design of a seven storey office building	Total energy use (↓) Heating demand (↓)	–	Non-geometric variables	Removing design variables (1-time Re-OPF)	NO	NO
	Shen and Tzempelikos [52]	Conceptual design of a one storey office space	Useful daylight illuminance (↑) Annual lighting, heating and cooling demand (↓) Annual source energy consumption (↓)	–	Window-to-wall ratio Space aspect ratio Non-geometric variables	Removing design variables (1-time Re-OPF)	NO	NO
Type 2	Trabelsi et al. [58]	Appliance scheduling in a smart home	Electricity cost (↓) Energy consumption (↓) Dissatisfied requests (↓) Budget for electricity cost Capacity of electric circuit Allowed time interval etc.	–	Starting time of multiple appliances	Modifying quantitative objective functions Modifying quantitative constraint values (4-time Re-OPF)	YES	NO
	Curtis et al. [59]	Conceptual design of a two-bar truss structure	Mass (↓) Deflection (↓) Stress Buckling stress etc.	–	Dimension of the structure Materials of the structure	Adding design variables (≥2-time Re-OPF)	YES	NO
	Curtis et al. [59]	Conceptual design of an aircraft	Cruise range (↑) Take-off weight (↓) Wetted aspect ratio Maximum lift to drag ratio Lift to drag ratio etc.	–	Wing aspect ratio	Adding and removing quantitative objectives (2-time Re-OPF) Adding design variables (4-time Re-OPF)	YES	NO
Type 3	Brintrup et al. [61]	Conceptual design of a one story plant layout	Cost of building (↓)	Subjective expert satisfaction (↑)	Dimensions of multiple rooms and areas	Adding qualitative objectives (1-time Re-OPF) Maintaining original quantitative objectives	NO	YES
	Mueller and Ochsendorf [62]	Conceptual design of a rigid frame structure	Use of material (↓)	Subjective aesthetic quality (↑)	3D coordinates of an inner profile of a rigid frame	Adding qualitative objectives (1-time Re-OPF) Maintaining original quantitative objectives	NO	YES
	Turrin et al. [66]	Conceptual design of a dome structure	Weight of structure (↓)	Subjective aesthetic quality (↑)	Geometry of the structure	Adding qualitative objectives (1-time Re-OPF) Maintaining original quantitative objectives	NO	YES
	Barnum and Mattson [67]	Conceptual design of a vehicle	Price (↓), Weight (↓), Seating (↓), Towing (↓), Cargo space (↓)	Subjective aesthetic quality (↑)	Geometry of the vehicle, Types of doors, chassis, engines, drive styles, cargo	Adding quantitative objectives (1-time Re-OPF) Maintaining original qualitative objectives	NO	YES
	Yang et al. [33]	Conceptual design of an indoor sports building	Useful daylight illuminance (↑) Daylight uniformity ratio (↑) Energy Use Intensity (↓) Structural mass (↓)	Subjective aesthetic quality (↑)	Geometry of the grandstand, building envelop, external shadings, roof structure	Modifying quantitative objective functions (1-time Re-OPF) Removing quantitative	NO	YES

(continued on next page)

Table 1 (*continued*)

Type	Literature	Application field	Initial quantitative objectives and constraints	Initial qualitative objectives and constraints	Initial design variables	Ways of Re-OPF	Dynamic Re-OPF *	Interactive Re-OPF *
			C-value, seat number, minimum space height Operative temperature Maximum utility, maximum displacement			objectives (1-time Re-OPF) Removing design variables (1-time Re-OPF) Maintaining original qualitative objectives		
Type 4	Newton [37]	Conceptual design of a solar shading façade	Useful daylight illuminance (↑) Condensation harvesting (↑)	–	Geometry of the façade	Adding and/or removing quantitative and qualitative objectives (2-time Re-OPF) Adding lateral-concept design variables (2-time Re-OPF)	YES	YES
	Kaushik and Janssen [69]	Conceptual design of an urban farm building layout	Compliance of adjacency rules (↑)	–	3D coordinates of spatial units	Adding and removing quantitative objectives (1-time Re-OPF) Adding and/or removing lateral-concept design variables (3-time Re-OPF)	YES	YES

and several final promising concepts are identified for eventual optimizations. The optimization results confirm the capability of the proposed method, and the usability of the techniques and tools.

1.4. Outline

The remainder of this paper is structured as follows. Section 2 reviews the state of the art of SBMOO methods, techniques and tools, and indicates gaps to be filled. Section 3 describes the overall procedure of the proposed method, and unfolds the dynamic and interactive Re-OPF phase. Section 4 provides more details on the adopted computational techniques. Section 5 provides more details on the adopted software tools. Section 6 presents a case study used to verify the capability of the proposed method and examine the usability of the techniques and tools. Finally, Section 7 summarizes contributions of this paper and some relevant aspects of the proposed method, and concludes with future research directions and concluding remarks.

2. Literature review

This section reviews the state of the art of SBMOO methods (in Section 2.1), and of SBMOO techniques and tools (in Section 2.2). The literatures presented here are not limited to those from the field of conceptual architectural design, due to the lack of relevant studies.

2.1. Review of SBMOO methods

This section reviews SBMOO methods which focus on Re-OPF, as shown in Table 1. Reviewing these methods relates to the necessity of Re-OPF for SBMOO methods. Such necessity has also been highlighted in [33]. In this section, first, four types of methods are introduced (in Section 2.1.1); then, examples of each type of methods are presented respectively (in Sections 2.1.2, 2.1.3, 2.1.4 and 2.1.5); finally, related gaps are identified (in Section 2.1.6).

2.1.1. Four types of methods

The methods under consideration can be classified into four types, according to whether or not Re-OPF is dynamic and interactive (Fig. 3). The four types of methods are: Type 1 methods with non-dynamic and non-interactive Re-OPF, Type 2 methods with dynamic and non-interactive Re-OPF, Type 3 methods with non-dynamic and interactive Re-OPF, and Type 4 methods with dynamic and interactive Re-OPF. This classification can help to understand the state of the art of the methods in allowing dynamic Re-OPF and/or interactive Re-OPF.

2.1.2. Type 1 methods with non-dynamic and non-interactive Re-OPF

Type 1 methods are characterized by non-dynamic and non-interactive Re-OPF (i.e., one-time Re-OPF where qualitative objectives are not considered, and lateral concept related variables are not revealed). The need of allowing such Re-OPF has been pointed out for nearly a decade in the aerospace and automotive industries. As stated in [49], it is necessary to include design space re-definition in optimization, given the fact that design requirements may change over time and significant re-designs can occur at a later time. Similarly, according to [50], allowing changing the set of design variables can allow new regions of the design space to be explored and lead to better designs.

Type 1 methods include some examples in relation to late-stage engineering design, for instance, those aiming at refining original design space using sensitivity analysis (e.g., [51,52]). In such examples, sensitivity analysis of multiple quantitative performance metrics to various design variables was conducted; then, the design variables were ranked according to their relative importance to each of the performance metrics. In this way, unimportant design variables were identified and screened out, thus refining the original design space. Given that design variables are removed once, but qualitative objectives are not considered, and lateral concept related variables are not revealed,

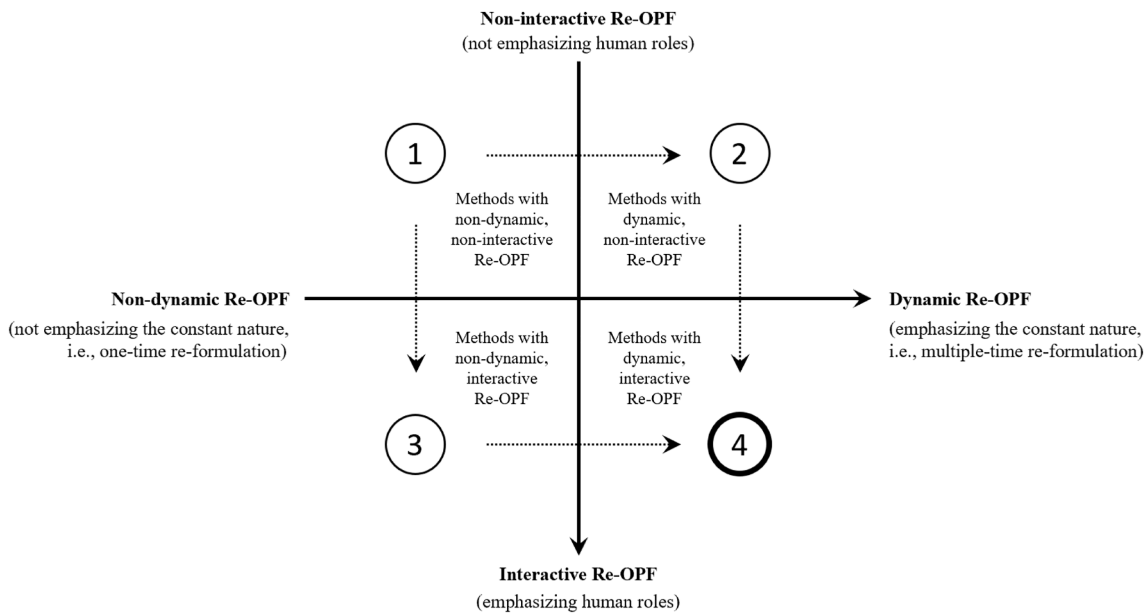


Fig. 3. Classification of SBMOO methods, according to whether or not Re-OPF is dynamic and interactive.

the above examples belong to Type 1 methods.

2.1.3. Type 2 methods with dynamic and non-interactive Re-OPF

Type 2 methods are characterized by dynamic and non-interactive Re-OPF (i.e., multiple-time Re-OPF where qualitative objectives are not considered, and lateral concept related variables are not revealed). The need of allowing such Re-OPF has been recognized recently in the building industry. Arora [53] stated that developing a proper formulation for a design optimization problem is an iterative process, namely, an initial formulation of the problem often needs several refinements or adjustments before an acceptable one is obtained.

Type 2 methods include "dynamic MOO" - a hot research topic in computer science [54–56], although it is seldom applied in conceptual architectural design. This topic aims at developing methods to solve dynamic MOO problems which involve time-varying objectives, constraints and design variables, such as control problems, scheduling problems, mechanical design problems, etc. [57]. To solve these problems, advanced dynamic MOO algorithms which have the ability to track the changing Pareto-optimal front are needed. They include at least two kinds: algorithms that solve dynamic MOO problems without adapting the problems, and algorithms that convert a dynamic MOO problem into multiple static MOO problems [57]. Trabelsi et al.'s example [58] was solved using the former algorithm; while Curtis et al.'s two examples [59] were solved using the latter algorithm. Curtis et al.'s examples also showed that the conversion or re-formulation of a dynamic MOO problem can be done by iteratively adding and removing objectives and design variables, and that this re-formulation can extend the exploration divergently into a larger space in order to avoid missing potentially superior solutions. Given that quantitative objectives and/or design variables are re-formulated multiple times, but qualitative objectives are not considered, and lateral concept related variables are not revealed, the above examples belong to Type 2 methods.

2.1.4. Type 3 methods with non-dynamic and interactive Re-OPF

Type 3 methods are characterized by non-dynamic and interactive Re-OPF (i.e., one-time Re-OPF where qualitative objectives are considered, and/or, lateral concept related variables are revealed). The need of allowing such Re-OPF has been recognized in the building industry. According to Cichocka et al.'s survey among architects [60], 91% of the surveyed architects would like to influence optimization outcomes by subjectively selecting promising designs, which indicates that human-in-the-loop methods seem appropriate in architectural design optimization. Brintrup et al. [61] claimed that a flexible optimization framework should be able to handle changing definitions of qualitative and quantitative criteria, constraints and criteria preferences. Mueller and Ochsendorf [62] stated that an ideal computational approach should expose designers to a diverse range of alternatives that may inspire new (quantitative and qualitative) goals or spark new (vertical and lateral) ideas.

Type 3 methods include "interactive evolutionary computation (IEC)" - a class of human-in-the-loop methods [63], although it is not often applied in conceptual architectural design. IEC relies on human subjectivity to evaluate qualitative performances that are normally difficult to be quantified explicitly [63]. For instance, human designers can quickly capture the value or beauty of buildings via observing their images [64,65]. IEC also promotes Re-OPF, it is extremely versatile in handling changing definitions of qualitative objectives as there is no need to hard-code qualitative influences [61]. Brintrup et al.'s example [61], Mueller and Ochsendorf's example [62], and Turrin et al.'s example [66] are among typical applications of IEC in conceptual architectural and structural design. In each of these examples, first, a single-objective optimization involving one quantitative objective was run; then, human designers were asked to subjectively evaluate qualitative performances of the obtained designs and select preferred designs for further optimization. In a certain circumstance (i.e., if there is no further optimization), the original optimization problem can be seen as being re-formulated once by adding a qualitative objective. Differently, Barnum and Mattson's example [67] showed a "reverse" method: first, a single-objective optimization involving one qualitative objective was run based on human subjective evaluation, in order to generate a quantitative preference-based model; then, a MOO problem was formulated by adding quantitative objectives, and thus run by using the preference-based model and other physics-based models. Moreover, instead of running optimization first, Yang et al.'s example [33] started with conducting a computational design exploration over a set of broad design samples preselected from the initial design space, with respect to the initial quantitative and qualitative objectives. During the exploration, numeric simulation was performed; quantitative and qualitative information were extracted via computational objective analysis of quantitative data and human subjective evaluation of qualitative data respectively; then, the two kinds of information were combined, in order to obtain comprehensive knowledge needed for re-formulating the initial MOO problem. Finally, optimization was run based on a re-formulated MOO problem. Given that objectives and/or design variables are re-formulated once, qualitative objectives are considered, the above examples belong to Type 3 methods.

2.1.5. Type 4 methods with dynamic and interactive Re-OPF

Type 4 methods are characterized by dynamic and interactive Re-OPF (i.e., multiple-time Re-OPF where qualitative objectives are considered, and/or, lateral concept related variables are revealed). The need of allowing such Re-OPF is seldom recognized in the building industry. Newton [37] identified major limitations of current MOO methods for architectural design, that is, the current methods do not accommodate flexible open-ended iterative design processes where design and objective spaces may change dynamically; they are not designed for finding novel and diverse designs; and they do not bring the designer into the loop in ways that stimulate the designer to be more creative. Actually, the first limitation indicates the need of dynamic Re-OPF, while the last two limitations indicate the need of interactive Re-OPF. Janssen [68] believed that it is helpful to have an adaptive - iterative design process which allows defining and re-defining a design search space, and thus shifting the boundaries of the space dynamically. Yang et al. [33] mentioned that the boundary shifting via "variable adding" should be encouraged in conceptual architectural design exploration, especially adding lateral concept related variables for more creative designs.

Type 4 methods have very few examples in relation to conceptual architectural design. Newton [37], Kaushik and Janssen [69] provide two valuable ones. In both examples, it is through dynamic and interactive Re-OPF phases that the design processes were driven forward, making the designs more complex and less abstract progressively. The dynamic and interactive Re-OPF phases are realized in different ways. In Newton's example, quantitative and qualitative objectives were added and/or removed two times, lateral concept related variables were added also two times; human designers are involved for handling qualitative objectives and devising lateral concept related variables. In Kaushik and Janssen's example, quantitative objectives were added once, lateral concept related variables were added and/or removed three times; human designers are involved for devising lateral concept related variables. Given that objectives and/or design variables are re-formulated multiple times, qualitative objectives are considered, and/or, lateral concept related variables are revealed, the above examples belong to Type 4 methods.

2.1.6. Gaps in SBMOO methods

In sum, there are a small number of SBMOO methods which focus on Re-OPF. Most of them do not allow dynamic Re-OPF and interactive Re-OPF simultaneously (i.e., Type 1, Type 2 and Type 3 methods). Only two of them allow dynamic and interactive Re-OPF to varying degrees (i.e., Type 4 methods). Although the two methods may have not been fully explored in some aspects, they do indicate the value of dynamic and interactive Re-OPF for conceptual architectural design exploration. Therefore, a SBMOO method which builds in a dynamic and interactive Re-OPF phase is suggested in this study, as elaborated in Section 3.

2.2. Review of SBMOO techniques and tools

This section reviews SBMOO tools each of which can be used to implement six predefined kinds of techniques, as shown in Table 2. In this study, a SBMOO tool refers to a combined software system consisting of individual software programs and/or customized tools. Reviewing these tools relates to their capabilities of supporting techniques needed for SBMOO methods which focus on Re-OPF. Such capabilities have been partially demonstrated in [33]. In this section, first, six kinds of techniques are introduced (in Section 2.2.1); then, examples of tools capable of supporting six kinds of techniques are presented (in Section 2.2.2); finally, related gaps are identified (in Section 2.2.3).

2.2.1. Six kinds of techniques

The tools under consideration can facilitate users to implement the following six kinds of techniques: geometric parametric modelling, multi-disciplinary simulation modelling, MOO, DoE sampling, quantitative data analysis, and qualitative data visualization. These kinds of techniques are used as relevant dimensions, based on which specific characteristics of the tools are presented. It should be clarified that tools supporting "non-geometric" parametric modelling and "mono-disciplinary" simulation modelling (e.g., MultiOpt [70], IDA-ICE + MATLAB [71], jEPlus + EA [72,73], MOBO [74], etc.) are out of scope of this review, as they are more suitable for late stages of detailed engineering design, rather than conceptual architectural design.

2.2.2. Tools capable of supporting the six kinds of techniques

The tools capable of supporting the above six kinds of techniques include early and recent generative design systems. In order to guide the readers to understand the authors' choice of the tool for this study, specific characteristics of these systems are presented below, based on the aforementioned six dimensions.

- Geometric parametric modelling

The reviewed systems may utilize different kinds of geometric parametric modelling techniques. Some systems apply non-visual programming, such as, a text-based programming language (as in [75–79], [80,81], [82,83]), a BIM technique (as in [84], [85,86]), or, a fast modelling technique (as in [87]). The other systems apply visual programming, such as, a visual programming language (as in [68,88,89], [90,91], [92,93], [94–97], [98], [33,47]).

- Multi-disciplinary simulation modelling

The reviewed systems may utilize different kinds of multi-disciplinary simulation modelling techniques with different integration capabilities. Some systems integrate only two kinds of simulations or calculations belonging to two disciplines (as in [75–79], [80,81], [82,83], [84], [85,86]). The other systems integrate more kinds of simulations or calculations belonging to more disciplines (as in [87], especially in [68,88,89], [90,91], [92,93], [94–97], [98], [33,47]).

- MOO

The reviewed systems may utilize different kinds of MOO algorithms based on different user interfaces. Some systems apply a MOO algorithm based on text-based user interfaces (as in [75–79], [80,81], [82,83]), or, user-friendly graphical user interfaces (as in [84], [85,86], [87], [90,91], [94–97]). The other systems have multiple advanced MOO algorithms to choose from based on user-friendly graphical user interfaces (as in [68,88,89], [92,93], especially in [98], [33,47]).

- DoE sampling

The reviewed systems may utilize different kinds of DoE sampling algorithms. Some systems apply a simple random sampling algorithm (as in [75–79], [80,81], [82,83], [84], [85,86], [68,88,89], [90,91], [92,93], [94–97]). The other systems have multiple advanced sampling algorithms to choose from (as in [87], especially in [98], [33,47]).

- Quantitative data analysis

The reviewed systems may utilize different kinds of quantitative data analysis techniques. Some systems apply only trade-off analysis (as in [75–79], [80,81], [82,83], [84], [85,86], [68,88,89], [90,91], [92,93]). The other systems have richer data analysis techniques to choose from (as in [87], [94–97], especially in [98], [33,47]).

Table 2
Review of SBMOO techniques and tools.

Literature	Tools	Six kinds of techniques (implemented using the tools)					
		(1) Geometric parametric modelling	(2) Multi-disciplinary simulation modelling	(3) MOO	(4) DoE sampling	(5) Quantitative data analysis	(6) Qualitative data visualization
Caldas [75–77] Caldas and Norford [78,79]	GENE_ARCH	Text-based programming (in Unix)	Energy simulation (in DOE-2.1E) Daylight simulation (in DOE-2.1E)	NSGA (in Unix)	Random sampling (in Unix)	Trade-off analysis (in –)	Separated visualization (in AutoCad, DrawDBL)
Wright et al. [80]	–	Text-based programming (in –)	Energy simulation (in EnergyPlus) Cost calculation (Customized)	NSGA-II (in –)	Random sampling (in –)	Trade-off analysis (in –)	Separated visualization (in –)
Shea et al. [81]	–	Text-based programming (in Matlab)	Daylight simulation (in Radiance) Cost calculation (Customized)	MACO (in Matlab)	Random sampling (in Matlab)	Trade-off analysis (in Matlab)	Combined visualization (in Matlab)
Conti [82] Conti et al. [83]	–	Text-based programming (in Processing)	Thermal calculation (Customized) View quality calculation (Customized)	NSGA-II (in Processing)	Random sampling (in Processing)	Trade-off analysis (in Processing)	Combined visualization (in Processing)
Gagne and Andersen [84]	–	BIM with limited parametric capabilities (in SketchUp)	Illuminance simulation (in Lightsolve Viewer) Glare simulation (in Lightsolve Viewer)	Micro-GA (in –)	Random sampling (in –)	Trade-off analysis (in –)	Separated visualization (in SketchUp)
Gerber and Lin [85,86]	H.D.S. Beagle (Revit plug-ins)	BIM with limited parametric capabilities (in Revit)	Energy simulation (in Green Building Studio) Cost calculation (Customized)	A GA-based MOO algorithm (in H.D.S. Beagle)	Random sampling (in H.D.S. Beagle)	Trade-off analysis (in –)	Separated visualization (in Revit)
DesignBuilder [87]	DesignBuilder	Fast modelling with limited parametric capabilities (in DesignBuilder)	Energy, thermal, carbon emission simulation (in EnergyPlus) Daylight simulation (in Radiance) Cost calculation (Customized)	NSGA-II (in DesignBuilder)	A few DoE sampling algorithms (in DesignBuilder)	Trade-off analysis Sensitivity analysis Uncertainty analysis (in DesignBuilder)	Separated visualization (in DesignBuilder)
Janssen et al. [68,88,89]	Dexen-Eddex	Visual programming (in Houdini)	Daylight simulation (in Houdarcs - Daysim) Energy, thermal simulation (in Houdarcs - EnergyPlus) Structure simulation (in Houdarcs – Calculix Z88) etc.	MOEAs, SIAs, customized algorithms (in Dexen-Eddex)	Random sampling (in Dexen-Eddex)	Trade-off analysis (in Dexen-Eddex)	Separated visualization (in Dexen-Eddex)
Von Buelow [90,91]	ParaGen	Visual programming (in GenerativeComponents, etc.)	Energy, thermal simulation (in Ecotect) Structure simulation (in STAAD- Pro) Acoustic simulation etc.	NDDP GA (in ParaGen)	Random sampling (in ParaGen)	Trade-off analysis (in ParaGen)	Combined visualization (in ParaGen)
Vierlinger and Bollinger [92] Negendahl and Nielsen [93]	Octopus.E (Grasshopper plug-ins)	Visual programming (in Grasshopper)	Daylight simulation (in Radiance) Energy simulation (in Be10), Thermal simulation (in HQSS) Cost calculation (Customized) etc.	SPEA2, HypE (in Octopus.E)	Random sampling (in Octopus.E)	Trade-off analysis (in Octopus.E)	Combined visualization (in Octopus.E)
Brown and Mueller [94–97]	DSE (Grasshopper plug-ins)	Visual programming (in Grasshopper)	Energy, thermal simulation (in Archsim - EnergyPlus) Structure simulation (in Karamba3D) etc.	NSGA-II (in DSE)	Random sampling (in DSE)	Trade-off analysis Sensitivity analysis Cluster analysis (in DSE)	Combined visualization (in DSE)

(continued on next page)

Table 2 (*continued*)

Literature	Tools	Six kinds of techniques (implemented using the tools)					
		(1) Geometric parametric modelling	(2) Multi-disciplinary simulation modelling	(3) MOO	(4) DoE sampling	(5) Quantitative data analysis	(6) Qualitative data visualization
Flager et al. [98]	–	Visual programming (in Digital Project etc.)	Energy, thermal simulation (in EnergyPlus) Structure simulation (in GSA) etc.	Darwin algorithm, NSGA-II, MOGA etc. (in ModelCenter)	Many DoE sampling algorithms (in ModelCenter)	Trade-off analysis Sensitivity analysis Probabilistic analysis, etc. (in ModelCenter)	Combined visualization (in ModelCenter)
Yang et al. [33,47]	GH-MF	Visual programming (in Grasshopper etc.)	Daylight simulation (in Ladybug, Honeybee - Daysim) Energy, thermal simulation (in Ladybug, Honeybee - EnergyPlus) Structure simulation (in Karamba3D) etc.	Many MOO algorithms (in modeFRONTIER)	Many DoE sampling algorithms (in modeFRONTIER)	Trade-off analysis Sensitivity analysis Cluster analysis Correlation analysis, etc. (in modeFRONTIER)	Combined visualization (in modeFRONTIER)

- Qualitative data visualization

The reviewed systems may utilize different kinds of qualitative data visualization techniques. Some systems apply separated visualization - showing 3D geometries and numeric data separately (as in [75–79], [80,84], [85,86], [87], [68,88,89]). The other systems apply combined visualization - showing 3D geometries and numeric data side-by-side simultaneously (as in [81], [82,83], [90,91], [92,93], [94–97], [98], [33,47]).

To facilitate the implementation of SBMOO methods which focus on Re-OPF, visual programming, broad simulation integration, advanced MOO algorithms, advanced sampling algorithms, rich data analysis, and combined visualization are all desired. Among these techniques, visual programming and rich data analysis are considered particularly important, due to the following facts. Visual programming can make geometric parametric modelling more user friendly for architects (many of whom may not have knowledge of text-based programming). Rich data analysis can serve as a major support for extracting information used for Re-OPF. In addition, the remaining techniques are also useful in other respects, for instance, in covering a wider range of performance requirements, improving search and sampling efficiency, and synthesizing information extracted from 3D geometries and numeric data.

Given the particular importance of visual programming and rich data analysis, tools to be used for this study should score high first on these two dimensions, while advantages on the other dimensions are also welcomed. Among the tools reviewed, generative design systems used in [98], [33,47] have relative advantages in the two important dimensions, as well as in the remaining dimensions. Thus, they are deemed more promising options for this study.

2.2.3. *Gaps in SBMOO techniques and tools*

In sum, visual programming and rich data analysis are particularly important techniques for supporting SBMOO methods which focus on Re-OPF; thus, the GH-MF system [33,47] which has relative advantages in these two techniques, is adopted as a promising option for this study. These two techniques face some challenges, for instance, visual programming requires proper flexibility of parametric models [99], rich data analysis requires proper use of analysis techniques. Besides, the GH-MF system needs more usability testing. To handle these challenges, some general solutions based on the GH-MF system have been suggested in a precedent study [33], but they are used for supporting a SBMOO method which involves non-dynamic Re-OPF. In contrast, some new and some previous solutions based on the GH-MF system are adopted in this study, so as to support a SBMOO method which involves dynamic Re-OPF. These solutions (i.e., techniques) and the GH-MF system adopted are elaborated in Section 4 and Section 5, respectively.

3. Proposed method

This section describes the overall procedure of the proposed method (in Section 3.1), and unfolds the dynamic and interactive Re-OPF phase (in Section 3.2).

3.1. *The overall procedure*

As shown in Fig. 4, the overall procedure of the proposed method consists of three phases: Initial-OPF, dynamic and interactive Re-OPF, OPS. These phases are respectively responsible for: formulating an initial MOO model; re-formulating previous MOO models; and solving a final re-formulated MOO model. The main innovation of the proposed method is the dynamic and interactive Re-OPF phase introduced between the Initial-OPF and OPS phases. Such Re-OPF phase is crucial for achieving a right or more realistic MOO problem and model before solving them. In the three phases of the overall procedure, several groups of actions (i.e., Action A-G) are appropriately arranged, as shown in Fig. 4 and described below.

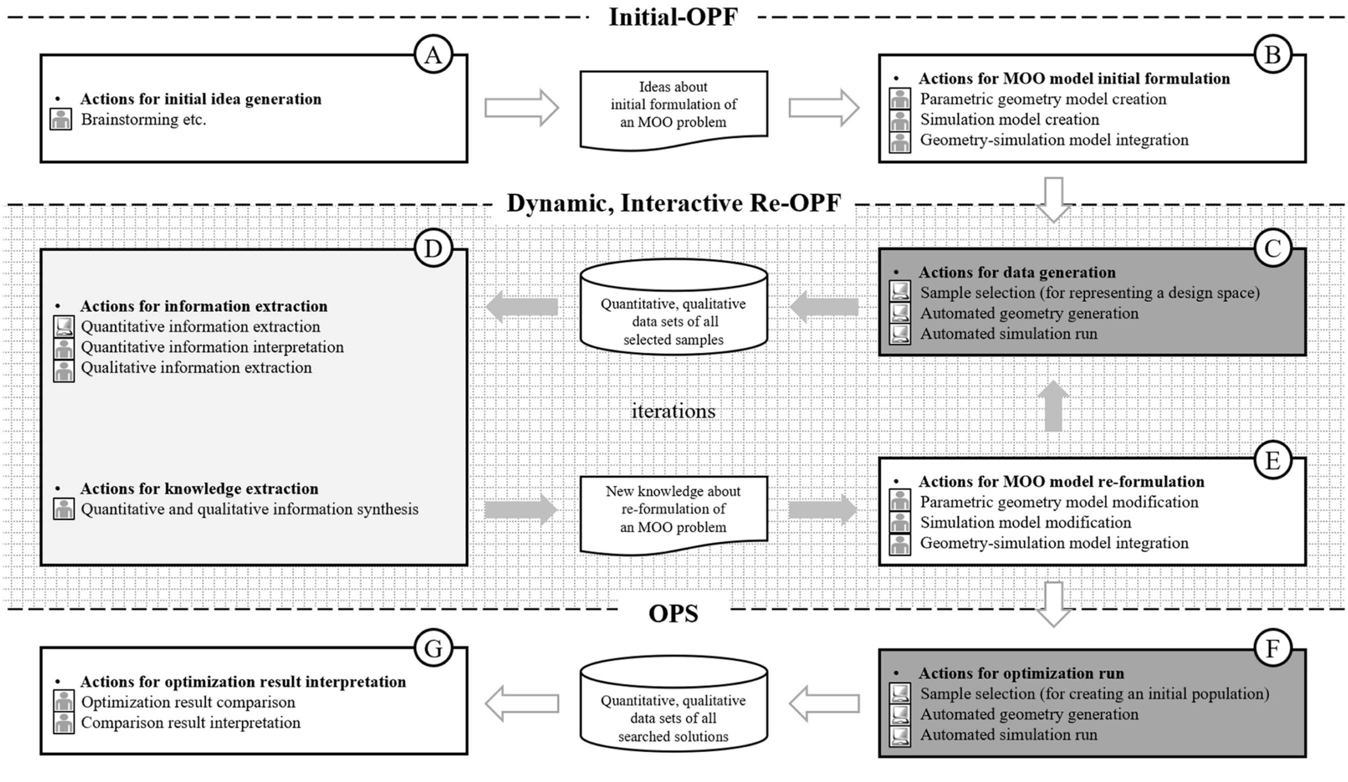


Fig. 4. The overall procedure of the proposed method. The dark gray, light gray, and white boxes show groups of actions mainly relying on computers, human-computer collaboration, and humans, respectively.

- Initial-OPF (Action A, B)

In initial idea generation (Action A), designers start with brainstorming ideas about initial formulation of an MOO problem (e.g., thoughts or suggestions about meaningful performance measures and objectives, promising design concepts and variables etc.). There can be multiple lateral concepts that are considered simultaneously, given the emphasis of divergent exploration.

In MOO model initial-formulation (Action B), an initial parametric geometry model and simulation models are created based on the initial ideas. Then, these models are integrated, to formulate an initial parametric geometry-simulation model which includes an initial set of objectives, constraints, and design variables (i.e., an initial MOO model).

- Re-OPF (Action C, D, E)

In data generation (Action C), a large set of samples are selected from the initial (or latest) design space, and used as a representation of the design space for exploring performance trends over the entire space. Based on the initial (or latest) parametric geometry-simulation model, the samples' 3D geometries are generated and their numerical simulations are run automatically. This automation is conducted in a sequential manner. Then, qualitative data sets (i.e., 3D geometries) and quantitative data sets (i.e., numeric design values, numerical simulation results) of all selected samples are collected.

In information and knowledge extraction (Action D), the quantitative data sets are analyzed and interpreted to acquire quantitative information; and the qualitative data sets are visualized to acquire qualitative information. Then, the two kinds of information are synthesized, to acquire comprehensive new knowledge about which performance measures and objectives, design concepts and variables can be added and/or removed.

In MOO model re-formulation (Action E), the initial (or latest) parametric geometry model and simulation models are modified based on the new knowledge. Then, the modified models are integrated, to formulate a new parametric geometry-simulation model which includes a new set of objectives, constraints, and design variables (i.e., a new MOO model).

At this point, designers can decide either to continue Re-OPF by iterating through the above three groups of actions (Action C, D, E), or to enter OPS. After the last Re-OPF iteration, a final parametric geometry-simulation model which includes a final set of objectives, constraints, and design variables (i.e., a final MOO model) is ready for use in the consequent OPS.

- OPS (Action F, G)

In optimization run (Action F), a small set of samples are selected from the final design space (specifically from final promising samples), and used as an initial population for searching optimal solutions within the design space. Based on the final parametric geometry-simulation model, searched solutions' 3D geometries are generated and their numerical simulations are run automatically. This automation is guided by an optimization algorithm. Then, qualitative and quantitative data sets of all searched solutions are collected.

In optimization result interpretation (Action G), the qualitative and quantitative data sets of optimal solutions are compared and interpreted, in order to acquire desired information and knowledge about the optimal solutions (e.g., trade-off relations etc.).

To sum up, the above actions are not arranged in a fixed and fully automated manner; instead, their arrangement provides necessary flexibility for the procedure to accommodate different needs. The flexibility is largely derived from the dynamic and interactive Re-OPF phase; thus, it is worth further understanding the mechanism of this particular phase, as elaborated in the next section.

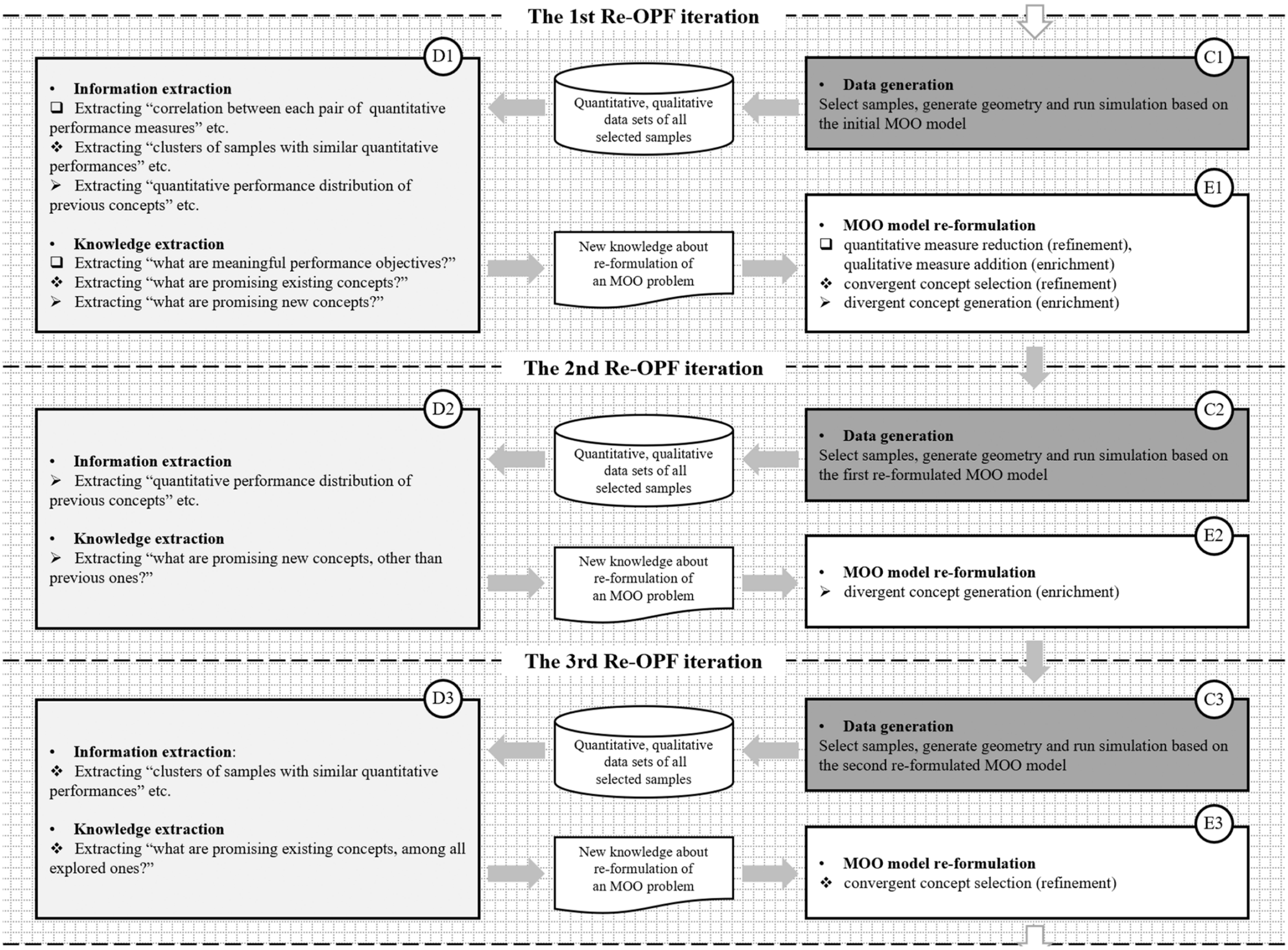


Fig. 5. A possible example of the Re-OPF phase which is unfolded to include three iterations.

3.2. The Re-OPF phase

As shown in Fig. 4, the Re-OPF phase consists of three groups of looped actions: data generation (Action C); information and knowledge extraction (Action D); and MOO model re-formulation (Action E). The Re-OPF phase has two characteristics: a dynamic Re-OPF characteristic - the above three groups of actions (Action C, D, E) are iterated in an open-ended manner; and, an interactive Re-OPF characteristic - the above three groups of actions (Action C, D, E) are performed by leveraging human capability and computation power in a collaborative manner.

The Re-OPF phase is flexible in, at least, two senses. It allows designers to include different numbers of Re-OPF iterations (within available time), and to perform actions of the same group in different ways (according to the desired extent of divergent exploration). In short, designers are allowed to customize the procedure of the Re-OPF phase with due flexibility. Here, a possible example of the Re-OPF phase (Fig. 5) is customized and used to explain the procedure. In this example, the Re-OPF phase is unfolded to include three iterations; actions of the same group (Action C1, C2, C3; Action D1, D2, D3; Action E1, E2, E3) are performed differently across the iterations, as shown in Fig. 5 and described below.

- The first Re-OPF iteration (Action C1, D1, E1)

In data generation (Action C1), data is generated based on the initial MOO model. In information and knowledge extraction (Action D1), three kinds of quantitative information are extracted, helping to extract three kinds of corresponding knowledge: "what are meaningful performance measures, what are promising existing concepts, and what are promising new concepts?". In MOO model re-formulation (Action E1), the extracted knowledge suggests to conduct quantitative measure reduction, qualitative measure addition, convergent concept selection, and divergent concept generation. By doing so, the first re-formulated MOO model is derived for the next Re-OPF iteration.

- The second Re-OPF iteration (Action C2, D2, E2)

In data generation (Action C2), data is generated based on the first re-formulated MOO model. In information and knowledge extraction (Action D2), one kind of quantitative information is extracted, helping to extract one kind of corresponding knowledge: "what are promising new concepts, other than previous ones?". In MOO model re-formulation (Action E2), the extracted knowledge suggests to conduct divergent concept generation. By doing so, the second re-formulated MOO model is derived for the next Re-OPF iteration.

- The third Re-OPF iteration (Action C3, D3, E3)

In data generation (Action C3), data is generated based on the second re-formulated MOO model. In information and knowledge extraction (Action D3), one kind of quantitative information is extracted, helping to extract one kind of corresponding knowledge: "what are promising existing concepts, among all explored ones?". In MOO model

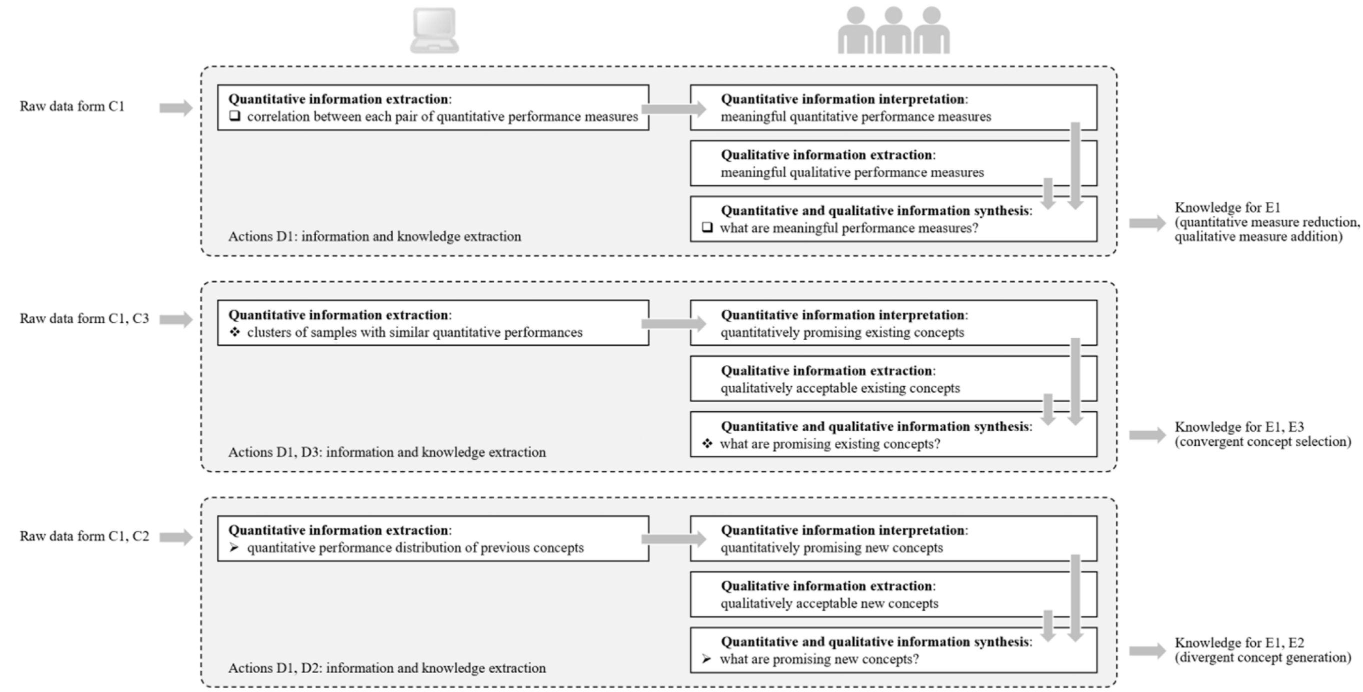


Fig. 6. Information and knowledge extraction relying on human-computer collaboration.

re-formulation (Action E3), the extracted knowledge suggests to conduct convergent concept selection. By doing so, the third re-formulated MOO model is derived for the consequent OPS.

To sum up, information and knowledge extraction (Action D1, D2, D3) can be seen as a crucial learning process where designers can improve their understanding on MOO problems. They can link data generation (Action C1, C2, C3) and MOO model re-formulation (Action E1, E2, E3), by converting raw data into useful knowledge based on human-computer collaboration (Fig. 6). Computers are responsible for quantitative information extraction; humans are responsible for quantitative information interpretation, qualitative information extraction, and information synthesis.

Data generation is the basis of information and knowledge extraction. In the above example, three sets of diverse design samples selected using sampling algorithms are used for generating performance data, rather than three sets of less diverse solutions searched or selected using optimization algorithms. This is mainly because the use of diverse design samples is in accordance with the emphasis of this paper - divergent exploration. Nevertheless, it is also possible to use optimization algorithms for data generation in other cases where divergent exploration is not so much emphasized.

MOO model re-formulation is the result of information and knowledge extraction. In the above example, the extracted knowledge suggests to conduct MOO model re-formulation in different ways for different Re-OPF iterations. Overall, divergent concept generation is emphasized here, which is conducted in the first and second Re-OPF iterations. If designers want to emphasize divergent concept generation at a higher level, they can increase the number of Re-OPF iterations involving it.

Table 3 shows actions of the Re-OPF phase (discussed in Section 3), adopted computational techniques and software tools (to be discussed in Section 4 and Section 5 respectively).

4. Computational techniques

This section provides more details on the adopted computational techniques. The techniques are used to implement the actions of the Re-OPF phase, as shown in Table 3. They are classified into three groups: techniques for data generation (in Section 4.1); techniques for information and knowledge extraction (in Section 4.2); and techniques for MOO model re-formulation (in Section 4.3).

4.1. *Techniques for data generation*

DoE sampling and tool integration techniques are used to implement data generation. In particular, uniform Latin hypercube sampling [100] as a DoE sampling technique (in Section 4.1.1) and custom system-to-system integration [101] as a tool integration technique (in Section 4.1.2), are respectively useful for selecting samples, automating geometry generation and simulation run.

4.1.1. *DoE sampling: uniform Latin hypercube sampling*

DoE sampling can help to get the maximum amount of information using the minimum amount of resources (i.e., a lower number of samples) [41,102]; thus, it is used to guide the choice of samples. The chosen samples represent an entire design space, in order to explore performance trends over the entire spectrum of the design space [98]. DoE sampling differs from an optimization technique, as it selects samples in a one-time manner before running all simulations, rather than selecting a small portion of samples at a time depending on simulation results of previous samples.

Uniform Latin hypercube sampling is a particular DoE sampling technique. It guarantees the lowest correlation between each pair of design variables and the highest uniform distribution [103]. Thus, samples well representing the entire design space can be selected using uniform Latin hypercube sampling.

4.1.2. *Tool integration: custom system-to-system integration*

Tool integration here not only refers to tool interoperability but also tool automation; thus, it is used to automate geometry generation and simulation run. Tool integration (automation) is an important requisite for working in a multidisciplinary design optimization framework [104]. It avoids users to click icons and enter data manually to perform tasks (e.g., geometry generation and simulation run) using graphic user interface [104]. In other words, it automates data flows between interconnected computer-aided design and computer-aided engineering

Table 3
Actions of the Re-OPF phase, computational techniques (used to implement the actions), and software tools (used to implement the techniques). Note: "*" marks techniques which are focused in Section 4.

Actions of the Re-OPF phase (Section 3)		Computational techniques (Section 4)		Software tools (Section 5)	
Types of actions	Specific actions	Types of techniques	Specific techniques	Types of tools	Specific tools
Data generation (Action C)	Sample selection	DoE sampling	Uniform Latin hypercube sampling *	Grasshopper (GH) modeFRONTIER (MF)	GH's slider components MF's DoE node
	Automated geometry generation Automated simulation run	Tool integration	Custom system-to-system integration *		GH's API MF's myNODE tool
Information and knowledge extraction (Action D)	Quantitative information extraction 1) extracting correlation between each pair of quantitative performance measures 2) extracting clusters of design samples with similar quantitative performances 3) extracting quantitative performance distribution of existing concepts	Quantitative data analysis 1) correlation analysis 2) cluster analysis 3) summary statistics	Self-Organizing Map (SOM) *	modeFRONTIER (MF)	MF's multivariate analysis tool (i.e., SOM creation tool)
			Hierarchical Clustering (HC) *		MF's multivariate analysis tool (i.e., HC creation tool)
			Box-whisker plot		MF's distribution analysis chart
	Quantitative information interpretation Qualitative information extraction Quantitative and qualitative information synthesis	Qualitative data visualization	Combined visualization		MF's run analysis interface (i.e., customizable visualization GUIs)
MOO model re-formulation (Action E)	Parametric geometry model modification	Geometric parametric modelling	Hierarchical variable structure *	Grasshopper (GH)	GH's Python script editor
			Modular programming *		GH's group and cluster features
	Simulation model modification Geometry-simulation model integration	Multi-disciplinary simulation modelling	Integrated dynamic models		GH's simulation plug-ins (e.g., Ladybug and Honeybee)

tools.

Custom system-to-system integration is a particular tool integration technique. It links generation, evaluation and selection tools in the same environment, usually for early design stages [101]. Typically, the generation and evaluation tools can be parametric design environments and their built-in simulation plug-ins; the selection tools can be process automation and optimization platforms. Thus, automated geometry generation and simulation run can be achieved using custom system-to-system integration.

4.2. Techniques for information and knowledge extraction

Quantitative data analysis and qualitative data visualization techniques are used to implement information and knowledge extraction. In particular, Self-Organizing Map (SOM) [105] as a correlation analysis technique (in Section 4.2.1), Hierarchical Clustering (HC) [106] as a cluster analysis technique (in Section 4.2.2), and box-whisker plots [102] as a technique to show summary statistics, are useful for extracting quantitative information. Combined visualization as a technique to simultaneously visualize numeric and non-numeric data, is especially useful for synthesizing quantitative and qualitative information. In practice, SOM and HC are relatively unfamiliar to designers, while box-whisker plots and combined visualization are more often used or easier to understand. Thus, the former two techniques are the focus of this section, rather than the latter ones.

4.2.1. Correlation analysis: self-organizing map

Correlation analysis can measure the strength of association between two variables and the direction of the relationship [42]; thus, it is used to extract "correlation between each pair of quantitative performance measures". Knowing such correlation and optimization goals of the measures (i.e., maximization or minimization), can help to identify meaningful quantitative performance measures from among possible ones. When two measures are positively and strongly correlated and their optimization goals are the same, or, when two measures are negatively and strongly correlated and their optimization goals are opposite, there are probably no meaningful trade-off relations between the two objectives. Thus, one of the measures can be considered as meaningful and kept, while the other one can be removed or treated as a constraint.

Self-organizing map is a particular correlation analysis technique. It is essentially an unsupervised neural network for ordering of high-dimensional data in such a way that similar data are grouped spatially close to one another [106]. Concisely, it is a dimensionality reduction method which can map multi-dimensional data into a two-dimensional space. It can be used to hunt for correlations [107,108], given the easy visualization and interpretation [109,110], as shown in Fig. 7 and described below.

As shown in Fig. 7 (left), a SOM (represented by a honeycomb-like diagram) is generated using a training data set and a learning algorithm. The training data set (marked by gray dots in the multi-dimensional data space) has quantitative performance measures as its dimensions (denoted by X, Y, Z). The learning algorithm is applied on the data set, in order to train prototype vectors (marked by black dots in the multi-dimensional data space). The prototype vectors' distribution approximates the probability density function of the training data [110]; each prototype vector corresponds to a group of similar training data. The obtained prototype vectors are projected onto a two-dimensional data space, forming a SOM.

As shown in Fig. 7 (right), the SOM of each quantitative performance measure is visualized and interpreted via a SOM plane. In the SOM plane, SOM units are colored according to related values of the

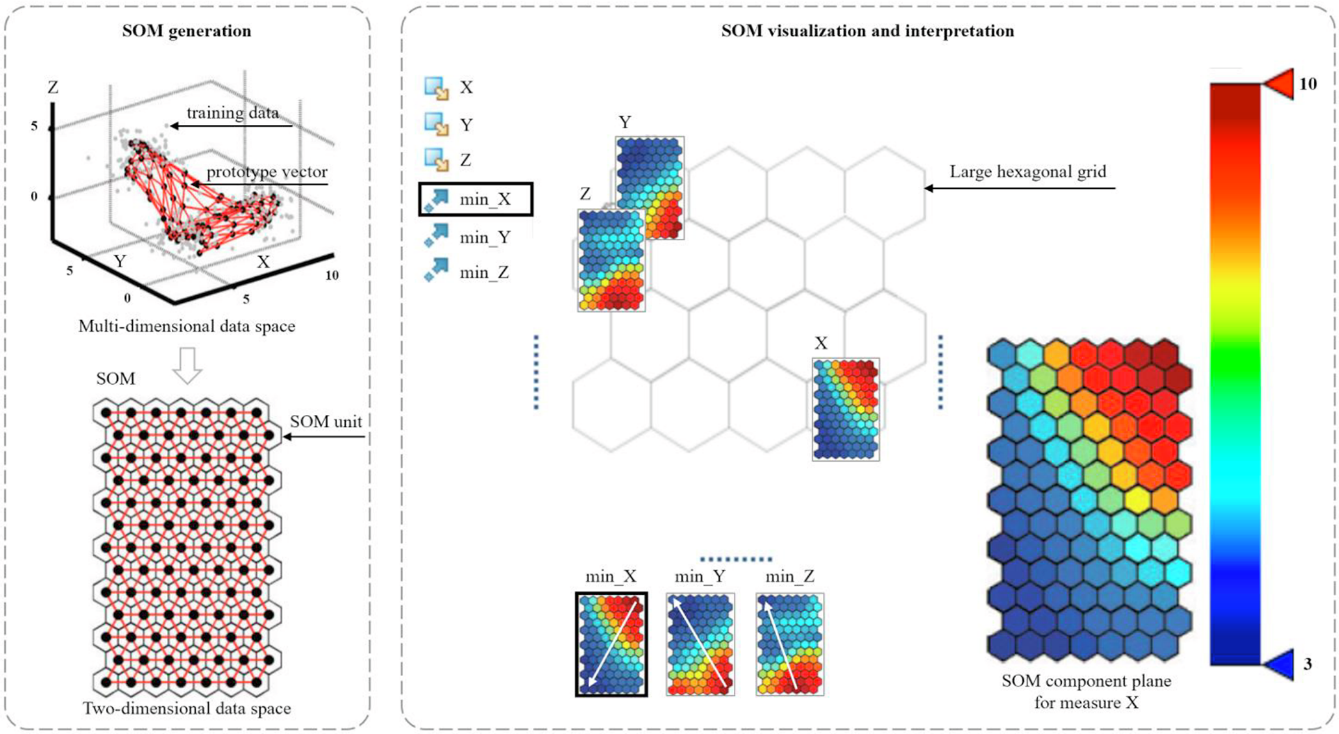


Fig. 7. Self-Organizing Map (SOM), revised from [110]. SOM generation (left); SOM visualization and interpretation (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

prototype vectors; the color scale runs from blue (i.e., low values) to red (i.e., high values). SOM planes for all measures are ordered on a large hexagonal grid, based on correlations between the measures. The more similar the color patterns and the closer the positions of the SOM planes, the stronger the correlations between the measures (e.g., Y and Z are positively and strongly correlated; X and Y are negatively and weakly correlated). This kind of pattern matching is something that human eye is very good at [108]. Moreover, arrows on top of SOM planes show the optimization goals of the measures. The arrows pointing from blue to red (i.e., from a low-value area to a high-value area) represent maximization goals; while the arrows with opposite directions represent minimization goals (e.g., X, Y, Z are all minimization goals). Overall, by observing the color patterns and arrows, correlations between each pair of quantitative performance measures and optimization goals of the measures can be easily understood, helping to identify meaningful quantitative performance measures.

4.2.2. Cluster analysis: hierarchical clustering

Cluster analysis can identify homogeneous clusters of samples in a source data set based on measured characteristics [102,106]; thus, it is used to extract "clusters of design samples with similar quantitative performances". Knowing such clusters, can help to identify quantitatively high-performing concepts from among existing ones. When a concept contains a relatively large number of samples belonging to quantitatively high-performing clusters, or in other words, when quantitatively high-performing clusters of samples mostly belong to a concept, this concept is more likely to produce desired solutions. Thus, this concept can be considered as quantitatively high-performing and kept for further exploration.

Hierarchical clustering is a particular cluster analysis technique. It is actually a versatile kind of approaches to clustering data, which produces a nested series of partitions rather than only one partition [43]. Concisely, it is a method to provide refined views to the inherent structure of the data. It can be used to group a large amount of data into manageable and meaningful clusters based on similarity, as shown in Fig. 8 and described below.

As shown in Fig. 8 (left), clusters (represented by a tree-like diagram called dendrogram) are generated using a source data set and a clustering algorithm. The source data set not only has quantitative performance measures (e.g., X, Y, Z) but also a high-level design variable (e.g., called "concept", to be mentioned in Section 4.3.1) as its dimensions. This facilitates to achieve clusters of design samples which have similar quantitative performances and belong to the same concept. The clustering algorithm specifies a linking method for building clusters. Based on the linking method, samples are merged, thus creating nested clusters (i.e., larger clusters created at later stages contain smaller clusters created at earlier stages) [106]. The dendrogram representing the nested clusters allows users to determine the number of clusters to be applied to the source data set. The number of intersections between the dash line and the dendrogram shows the number of clusters applied.

As shown in Fig. 8 (right), the clusters applied are visualized and interpreted using a clustering parallel coordinate chart. The chart creates a colored band for each of the clusters. In the colored band, the intersections between the thick center polyline and the parallel vertical lines represent the means of the quantitative performance measures and the high-level design variable; the band width represents the confidence intervals of the means. The green cluster consists of design samples from concept 1 and 2, as the mean of the high-level design variable "concept" is between 1 and 2. Arrows along the vertical lines of the quantitative performance measures reflect human preference on relative importance of the measures. The closer an arrow reaches the desired bound of the measure, the more important the measure is. Overall, by "pushing" the arrows to desired directions in desired extents, clusters of design samples with high quantitative performances can be quickly found, helping to identify quantitatively high-performing concepts.

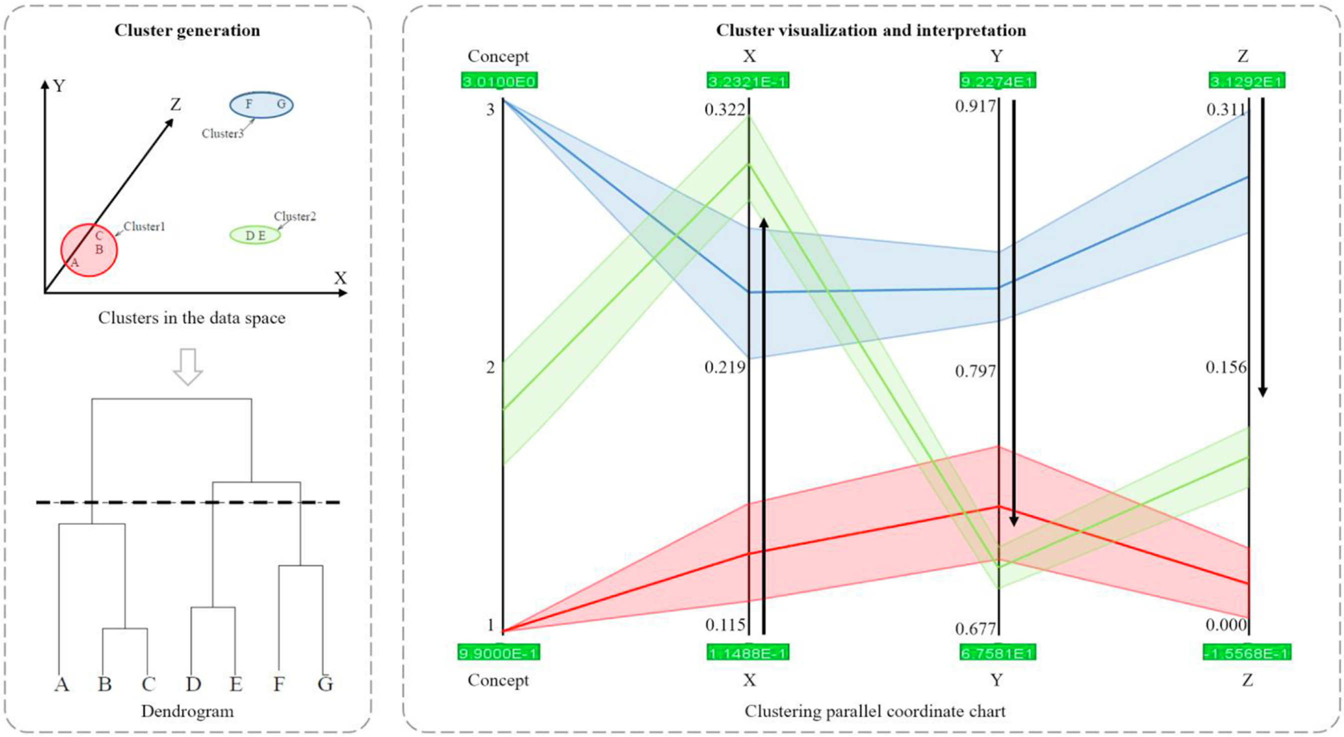


Fig. 8. Hierarchical Clustering (HC), revised from [43]. Cluster generation (left); cluster visualization and interpretation (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Techniques for MOO model re-formulation

Geometric parametric modelling, multi-disciplinary simulation modelling techniques are used to implement MOO model re-formulation. In particular, hierarchical variable structure [111] (in Section 4.3.1) and modular programming [112,113] (in Section 4.3.2) as geometric parametric modelling techniques, are useful for modifying parametric geometry models in a more flexible manner. Integrated dynamic model [5] as a multi-disciplinary simulation modelling technique, is useful for modifying simulation models and integrating geometry-simulation models. In practice, hierarchical variable structure and modular programming are relatively unfamiliar to designers, while integrated dynamic model (i.e., a combination of a design tool, a visual programming language and building performance simulation tools [5]) has become often used. Thus, the former two techniques are the focus of this section, rather than the latter one.

4.3.1. Geometric parametric modelling: hierarchical variable structure

Geometric parametric modelling should be used in a flexible way which allows the inclusion of different sets of lateral concept related variables in a parametric model. This flexibility is especially meaningful when many lateral geometric concepts need to be considered parametrically, as in this paper.

Hierarchical variable structure is a particular parametric modelling technique that can help to realize such flexibility. It often exists in product design in which a number of substructures and parts are hierarchically assembled into a larger system [111]. In this context, design variables may be from different levels of the hierarchy; and naturally, they are organized using a hierarchical structure, rather than a flat, one-dimensional array structure. This hierarchical structure consists of high-level and low-level variables. The values of high-level variables determine the selection of low-level variables; thus, the dimensionality of the resulting design space is changeable.

In this study, a two-level hierarchical variable structure is used, as shown in Fig. 9 (top left and bottom). The high-level variable (i.e., input variable 0, called “concept”) represents the type of design concepts; the low-level design variables (i.e., the remaining input variables) include those necessary to define the concepts.

4.3.2. Geometric parametric modelling: modular programming

Geometric parametric modelling should be used in a flexible way which facilitates the modification of parametric schemata for different lateral concepts. This flexibility is especially meaningful when many lateral geometric concepts need to be considered parametrically, as in this paper.

Modular programming is a particular parametric modelling technique that can help to realize such flexibility. It structures parametric schemata into modules. As defined in [112], a module in a dataflow programming language is a sequence of program instructions bounded by an entry and exit point, which performs a particular task. The entry point(s) collects data the module requires; the exit point(s) returns data the module produces; and the program instructions in between can be evoked by passing data through the module. According to [113], parametric schemata structured with modular programming principles are consistently better understood, particularly when the parametric model is complex and used in a collaborative environment.

In this study, the parametric schemata are structured into geometry generation modules and performance simulation modules, as shown in Fig. 9 (top middle and top right). Each geometry generation module corresponds to a group of geometric variations belonging to a particular design concept; each performance simulation module corresponds to a particular type of simulation.

5. Software tools

This section provides more details on the adopted software tools. The tools are used to implement the techniques needed for the Re-OPF phase, as shown in Table 3. The purposes of choosing the tools are introduced (in Section 5.1), and the integration of the tools is described (in Section 5.2).

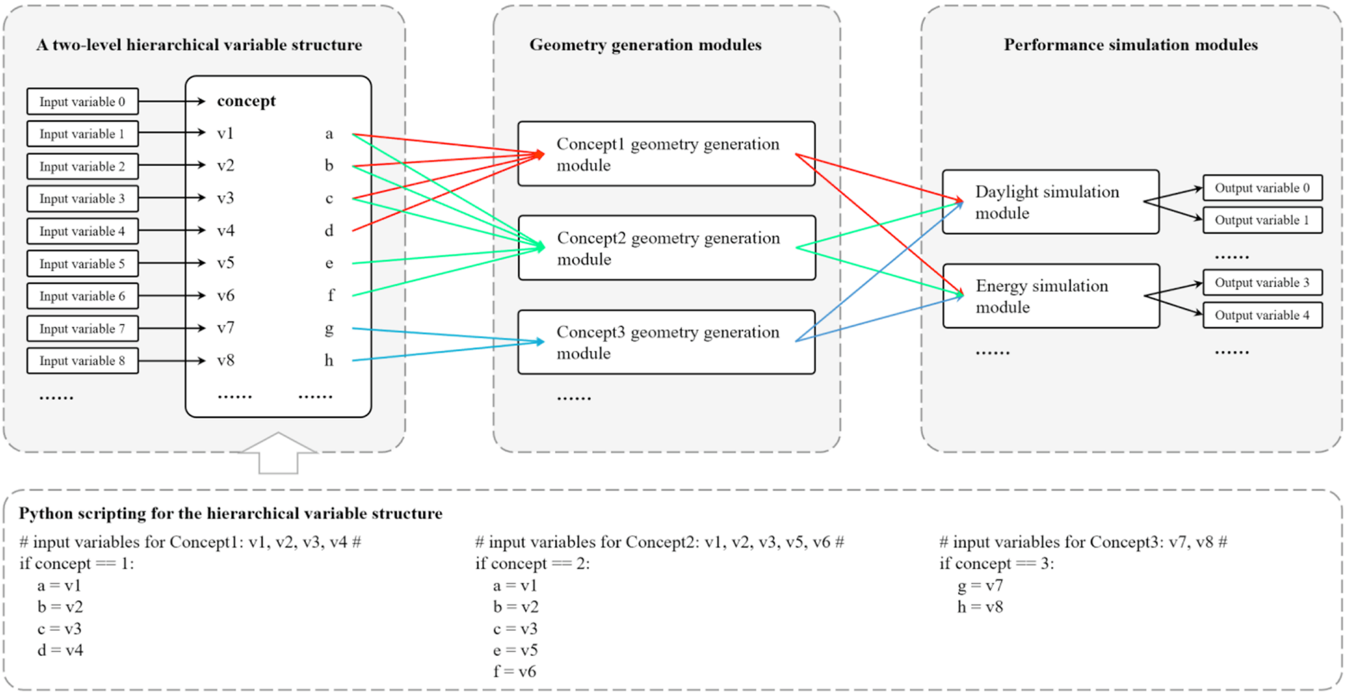


Fig. 9. The overall structure of parametric schemata. A two-level hierarchical variable structure (top left); geometry generation modules (top middle); performance simulation modules (top right); Python scripting for the hierarchical variable structure (bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.1. Choice of Grasshopper and modeFRONTIER

McNeel's Grasshopper [45], is a visual programming environment for the 3D modeler Rhinoceros. It is the most popular parametric design tool among architectural design professionals, due to its intuitive way of exploring geometries without having to know scripting [60]. Grasshopper and its plug-ins are chosen to implement geometric parametric modelling, multi-disciplinary simulation modelling techniques (mentioned in Section 4.3). Specifically, hierarchical variable structure and modular programming can be implemented using Grasshopper's Python script editor, group and cluster features, respectively; integrated dynamic model can be implemented using Grasshopper's simulation plug-ins (e.g., Ladybug and Honeybee [114] for linking simulation engines Daysim [115] and EnergyPlus [116]).

ESTECO's modeFRONTIER [46], is a process automation and optimization platform. It is an often-used multi-disciplinary engineering design exploration tool. modeFRONTIER is chosen to implement quantitative data analysis and qualitative data visualization techniques (mentioned in Section 4.2). Specifically, self-organizing map and hierarchical clustering can be implemented using modeFRONTIER's multivariate analysis tools; box-whisker plots can be implemented using modeFRONTIER's distribution analysis chart; combined visualization can be implemented using modeFRONTIER's run analysis interface.

Grasshopper and modeFRONTIER are chosen also for their capabilities of supporting DoE sampling and tool integration techniques (mentioned in Section 4.1). Specifically, uniform Latin hypercube sampling can be implemented using modeFRONTIER's DoE node and Grasshopper's slider components; custom system-to-system integration can be implemented using modeFRONTIER's myNODE tool and Grasshopper's API.

It is worth noting that Grasshopper and modeFRONTIER are not the only choices for supporting the aforementioned techniques. Other visual programming environments that could be chosen include Bentley's GenerativeComponents [117], Autodesk's Dynamo Studio [118], Gehry Technologies' Digital Project [119], Sidefx' Houdini [120], etc. Other process automation and optimization platforms that could be chosen include Phoenix Integration's ModelCenter [121] etc. The combination of Grasshopper and modeFRONTIER is one available option among others, which is adopted in this study.

5.2. Integration of Grasshopper and modeFRONTIER

The chosen tools Grasshopper and modeFRONTIER need to be integrated, in order to form a promising SBMOO tool GH-MF. In fact, the GH-MF integration (Fig. 10) is a form of custom system-to-system integration, and facilitates the implementation of other related techniques. It refers to the interoperability and automation between GH and MF, as shown in Fig. 10. In such integration, data exchange between GH and MF is automated. That is, modeFRONTIER automatically sends numeric input data to drive geometry generation and simulation run; and, Grasshopper automatically returns numeric output data to initiate the next iteration. This automatic data exchange repeats for all pre-selected samples. All the data, including numeric and non-numeric data, are stored in a database for later analysis and visualization.

The GH-MF integration is realized through a “GH-MF node”. The GH-MF node is a custom integration plug-in for modeFRONTIER, which is developed using modeFRONTIER's myNODE tool and Grasshopper's API. Specifically, the myNODE tool packages integration scripts into a myNODE file that can be installed in modeFRONTIER; once installed, the GH-MF node is created. There are different versions of the GH-MF node, which have been applied in previous studies [33,47,122,123]. The version used in this study is updated, and the same as that in [33]. Compared to the older versions in [47,122,123], this updated version can significantly streamline the integration process and has more advantages. Specifically, through the use of Grasshopper's API, the updated GH-MF node can automatically recognize and propagate input and output variables from GH to MF; and, it enables direct data exchange between GH and MF, without the need to specify external templates. Moreover, the updated GH-MF node enables one-click initiation of simulation run, rather than sequential clicking in both GH and MF (which may cause connection issues); and, it improves the stability of simulation run by calling and closing Grasshopper and

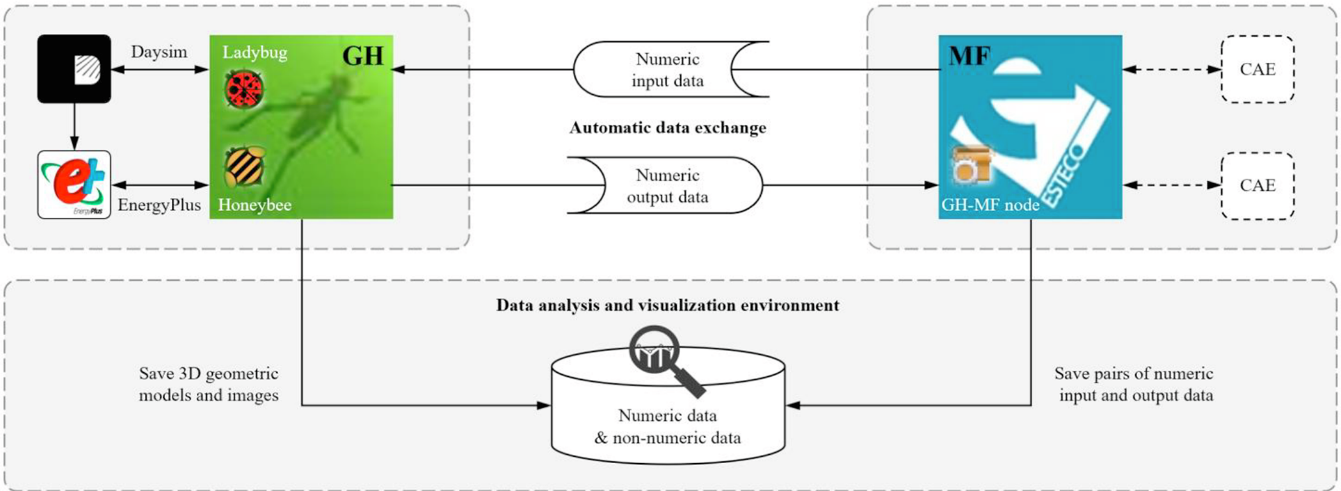


Fig. 10. The GH-MF integration. Grasshopper and its simulation plug-ins: Ladybug and Honeybee (top left); modeFRONTIER and its integration plug-in: GH-MF node (top right); a database for storing numeric and non-numeric data (bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Rhinoceros automatically for each iteration, rather than keeping them always alive (which may increase the risk of crashing).

6. Case study

This section presents a case study used to verify the capability of the proposed method and examine the usability of the techniques and tools. The case study is about the conceptual design of the top-daylighting system of a 40m × 70m × 15m indoor sports hall. It is carried out according to the overall procedure of the proposed method (Fig. 4) in which the Re-OPF phase includes three iterations (Fig. 5). First, the initial-OPF phase is described (in Section 6.1), where the initial MOO model is formulated. Then, three Re-OPF iterations are followed (in Section 6.2, 6.3 and 6.4 respectively), where previous MOO models are re-formulated. Last, the OPS phase is presented (in Section 6.5), where optimizations are conducted based on final re-formulated MOO models. Note that, in each Re-OPF iteration, the information and knowledge extraction (Fig. 6) are focused, given their importance.

6.1. Initial-OPF (Action A, B)

6.1.1. Initial idea generation (Action A)

Top daylighting is an effective way of bringing natural lights deep into buildings; thus, it is often used in large space like indoor sports halls. The initial idea of this case is to divergently explore the geometries of three typical types of top-daylighting concepts, in order to fulfill daylight, energy, cost and aesthetic performance requirements. Based on this idea, the initial MOO model is formulated as below.

6.1.2. MOO model initial-formulation (Action B)

- Initial design concepts and variables

Three typical types of top-daylighting concepts are considered as initial concepts, as shown in Fig. 11 (top). They are: Concept 1_0 (i.e., skylights), Concept 2_0 (i.e., roof monitors), Concept 3_0 (i.e., saw-tooth roofs) [48]. The initial variables for these concepts are organized in a two-level variable structure. A high-level variable is used to represent the type of the concepts; and, three different sets of low-level variables are used to define the geometries of the concepts. That is, when the value of the high-level variable "concept" is given, a particular set of low-level variables is chosen automatically to define the geometries of the given concept. A parametric geometry model is created using these initial variables.

In this paper, the focus is not on highlighting the complexity of geometries, but rather on showing how to continually enrich multiple concepts in a more informed manner. Thus, initial design variables in different cases may be more complex or less complex, depending on designer's preference, design contexts, etc.

- Initial performance requirements and measures

Four kinds of quantitative performance requirements from different disciplines are considered as initial requirements, as shown in Table 4. They are: energy use, daylight availability, daylight uniformity, and investment cost for glass. Some possible initial performance measures, and the associated goals, abbreviations and definitions are also shown in Table 4. Energy and daylight simulation models are created for these measures. In the energy simulation model, the weather file of Guangzhou is used; in the daylight simulation model, 66 illuminance test points evenly spread over the indoor space are used. These simulation models are integrated with the parametric geometry model, to form the initial MOO model.

In this paper, the focus is not on discussing the completeness of the initial performance measures, but rather on showing how to find meaningful performance measures in a more informed manner. Thus, initial performance measures in different cases may be more complete or less complete, depending on designer's prior knowledge, design contexts, etc.

6.2. The first Re-OPF iteration (Action C1, D1, E1)

6.2.1. Data generation (Action C1)

The first 300 data sets are generated automatically, based on an initial automation workflow. The workflow is established by uploading the initial MOO model, specifying the uniform Latin hypercube sampling algorithm and a sequential execution order. The data derived includes quantitative data (i.e., numeric design values, numeric simulation values) and qualitative data (i.e., 3D geometries), as shown in Fig. 12.

6.2.2. Information and knowledge extraction (Action D1)

As shown in Fig. 5, three kinds of knowledge are extracted during information and knowledge extraction (Action D1), as summarized below.

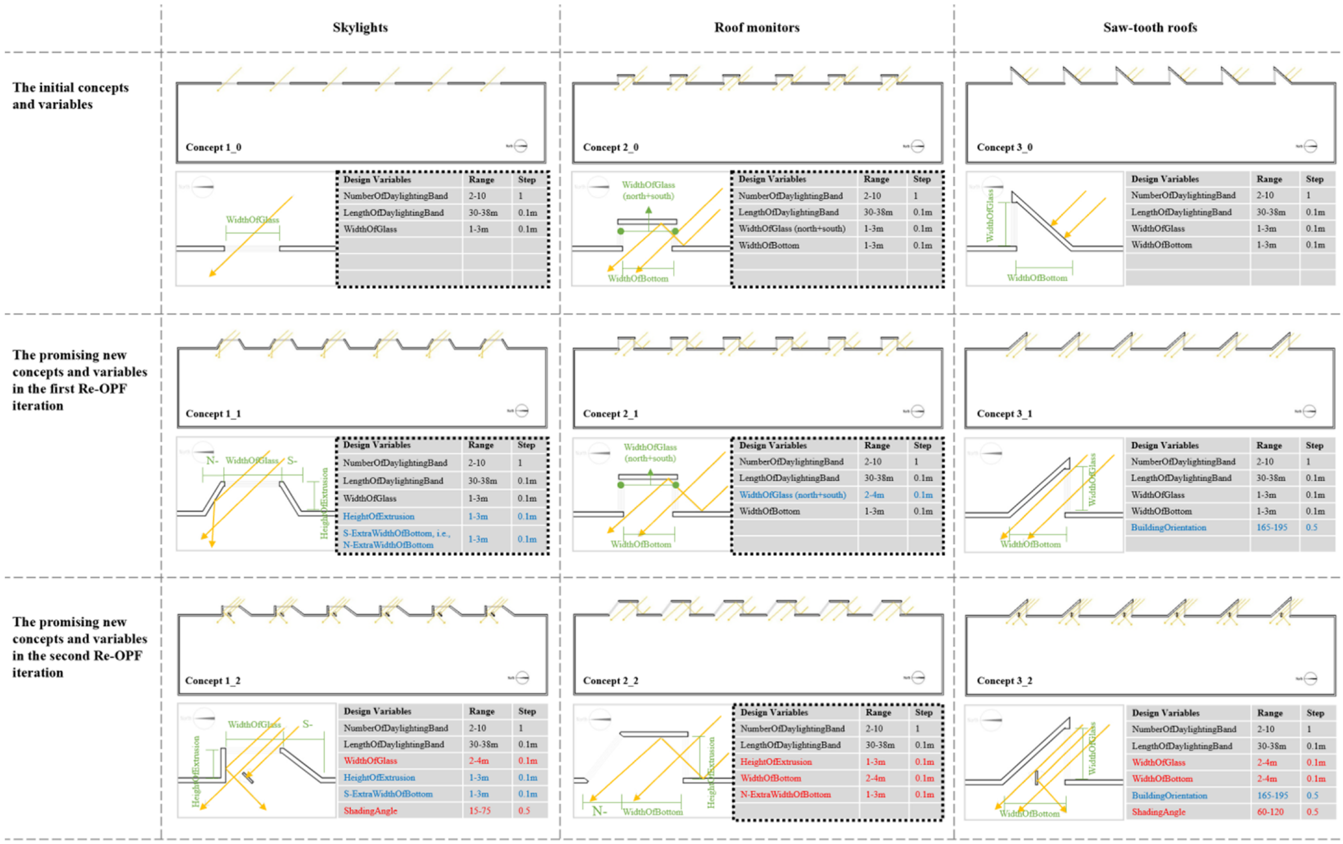


Fig. 11. All concepts and variables considered in different phases of the case study. The dashed boxes show the variables of the final promising existing concepts in the third Re-OPF iteration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Extracting knowledge of meaningful performance measures

To find meaningful quantitative performance measures, correlation between each pair of quantitative performance measures is extracted from the data, and interpreted by humans (Fig. 6). The quantitative data (having all the initial measures as its dimensions) is analyzed using self-organizing map. As a result, SOM planes for all the initial measures are generated (Fig. 13, left). Via human interpretation, three SOM

Table 4
Initial performance measures and related goals, abbreviations and definitions.

Category	Performance measures	Goals	Abbreviations	Definitions
Energy use	Energy Use Intensity	Minimization	EUI	Annual energy use per square meter of floor area
	Percentages of Cooling	Minimization	PoC	Percentages of energy use for cooling, heating, lighting and equipment respectively (which can be meaningful objectives, if they account for major portions of energy use)
	Percentages of Heating	Minimization	PoH	
	Percentages of Lighting	Minimization	PoL	
	Percentages of Equipment	Minimization	PoE	
Daylight availability	Useless Daylight Illuminance (< 100)	Minimization	UDI (< 100)	Percentage of floor area that meets the specified illuminance range for at least 50% of the occupied time
	Useful Daylight Illuminance (100–2000)	Maximization	UDI (100–2000)	
	Useless Daylight Illuminance (> 2000)	Minimization	UDI (> 2000)	
	Day Lit Area	Maximization	DLA	Percentage of floor area that receives illuminances above 300 lx for at least 50% of the occupied time
	Over Lit Area	Minimization	OLA	Percentage of floor area that receives illuminances above 3000 lx for at least 5% of the occupied time
Daylight uniformity	Average Uniformity	Maximization	AU	Annual average of illuminance uniformity ratios
Investment cost for glass	Area of Glass	Minimization	AoG	Total area of the glass used for top windows

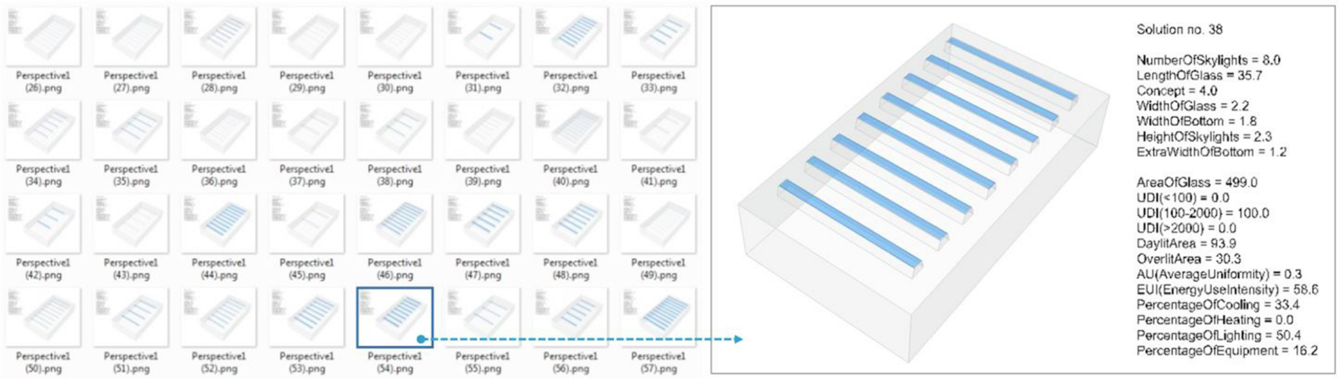


Fig. 12. Examples of building geometries, numeric design values, and numeric simulation values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

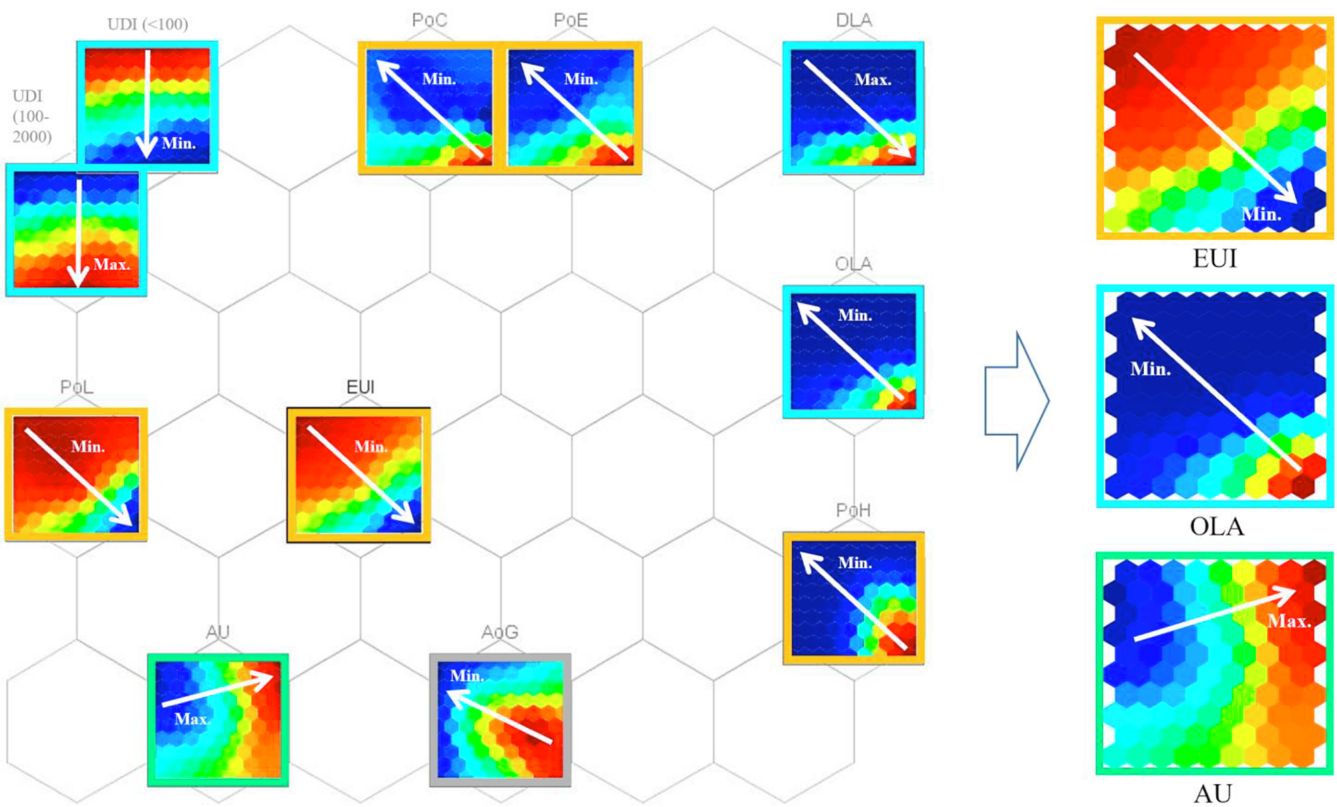


Fig. 13. SOM planes. SOM planes for all the initial measures (left); SOM planes for three meaningful quantitative performance measures (right). The yellow, blue, green and grey boxed lines, show SOM planes for energy use, daylight availability, daylight uniformity and investment measures, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

planes are identified (Fig. 13, right), indicating three meaningful quantitative performance measures, as summarized below.

(1) PoL is a meaningful quantitative measure, as lighting energy use dominates total energy use in this case. EUI is also a meaningful quantitative measure, as it facilitates direct comparison of energy use among different buildings. Either of them can be chosen, given that they are positively and strongly correlated and their optimization goals are the same. In this case, EUI is chosen (to form a minimization objective).
(2) For UDI (< 100) and UDI (100-2000), either of them can be chosen, given that they are negatively and strongly correlated and their optimization goals are opposite. For OLA and DLA, DLA is redundant with EUI, given that they are negatively correlated and have opposite optimization goals; OLA is more meaningful, which not only reflects daylight availability but also the risk of glare or overheating. In this case, OLA is chosen (to form a minimization objective).
(3) For AU and AoG, both of them can be considered as meaningful quantitative measures, given that they (especially AU) have weak correlation with the above chosen measures. In this case, AU is chosen (to form a maximization objective).

To find meaningful qualitative performance measures, human subjectivity is needed. Aesthetics is considered as a meaningful qualitative measure and chosen (to form a constraint). Overall, EUI, OLA, AU and Aesthetics are chosen as meaningful performance measures.

- Extracting knowledge of promising existing concepts

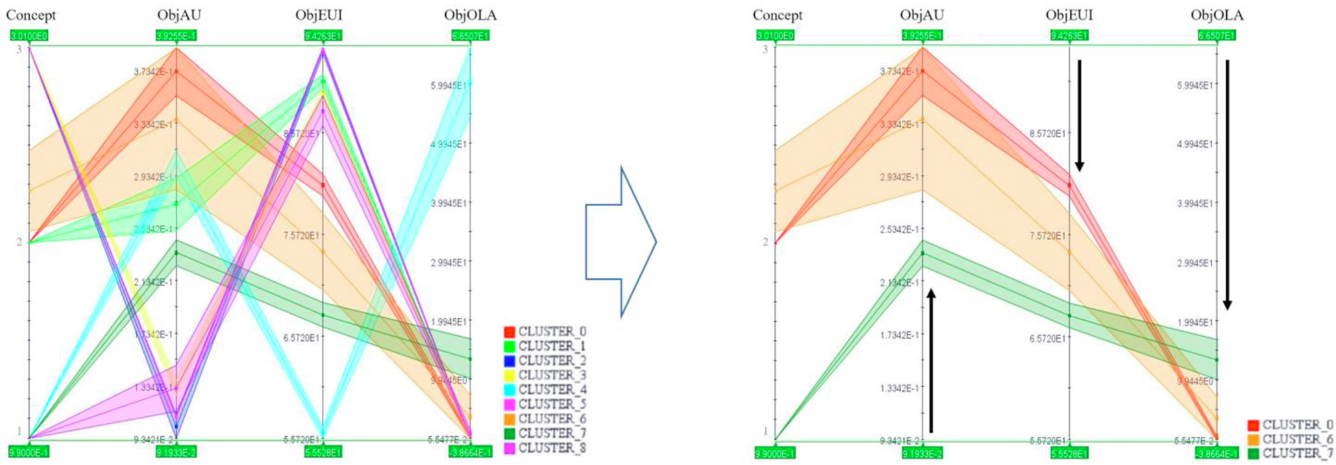


Fig. 14. Clusters in the first Re-OPF iteration. Nine clusters of samples generated (left); three clusters of samples identified (right). CLUSTER_0 consists of samples from Concept 2_0; CLUSTER_6 consists of samples from Concept 2_0 and Concept 3_0; CLUSTER_7 consists of samples from Concept 1_0. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To find quantitatively promising existing concepts, clusters of samples with similar quantitative performances are extracted from the data, and interpreted by humans (Fig. 6). The quantitative data (having the chosen measures EUI, OLA, AU and the high-level variable "concept" as its dimensions) is analyzed using hierarchical clustering. As a result, nine clusters of samples are generated (Fig. 14, left). Via human interpretation, three clusters are identified (Fig. 14, right), indicating quantitatively promising existing concepts, as summarized below.

(1) Concept 1_0 and Concept 2_0 are quantitatively promising existing concepts, as the identified clusters of samples mostly belong to these two concepts.
(2) Concept 3_0 is a less quantitatively promising existing concept, as the identified clusters of samples mostly do not belong to this concept (i.e., very few samples in the identified CLUSTER_6 are from Concept 3_0, as shown in Fig. 14).

To judge qualitatively acceptable existing concepts, human subjectivity is needed. Concept 1_0, Concept 2_0 and Concept 3_0 are all judged as aesthetically acceptable. Overall, they are chosen as promising existing concepts (although Concept 3_0 is less quantitatively promising).

- Extracting knowledge of promising new concepts

To find quantitatively promising new concepts, quantitative performance distribution of previous concepts is extracted from the data, and interpreted by humans (Fig. 6). The quantitative data (having the chosen measures EUI, OLA, AU as its dimensions) is summarized using box-whisker plots and scatter plots. As a result, such plots for EUI, OLA, AU of the previous concepts (i.e., Concept 1_0, Concept 2_0 and Concept 3_0) are generated (Fig. 15). Via human interpretation, room for possible improvements is found, indicating new design strategies and concepts, as summarized below.

(1) For Concept 1_0, there is room for improving OLA and AU while maintaining EUI. Reducing direct sunlight and introducing reflected daylight can be respectively helpful for reducing OLA and increasing AU. Allowing daylight to enter without obstacles through the top-facing window is meaningful for maintaining EUI. Based on these strategies, Concept 1_1 is produced which creates inclined opaque elements by lifting the skylight and expanding the opening on the roof bottom surface.
(2) For Concept 2_0, there is room for improving EUI while maintaining OLA and AU. Introducing more daylight into the space can be helpful for reducing EUI. Blocking out some daylight from high angles with the horizontal protruding opaque element is meaningful for maintaining OLA and AU. Based on these strategies, Concept 2_1 is produced which enlarges the window size by lifting the protruding element.
(3) For Concept 3_0, there is room for improving EUI and AU while maintaining OLA. Introducing a proper amount of daylight from near the south can be helpful for reducing EUI and increasing AU. Blocking out some daylight from the opposite side of the window with the inclined protruding opaque element is meaningful for maintaining OLA. Based on these strategies, Concept 3_1 is produced which changes the orientation of the window around the south.

To judge qualitatively acceptable new concepts, human subjectivity is needed. Concept 1_1, Concept 2_1 and Concept 3_1 are all judged as aesthetically acceptable. Overall, they are chosen as promising new concepts.

6.2.3. MOO model re-formulation (Action E1)

The above knowledge suggests to conduct: quantitative measure reduction and qualitative measure addition, convergent concept selection and divergent concept generation. Specifically, the initial energy and daylight simulation models are modified, by reducing some initial quantitative measures; the initial parametric geometry model is modified, by using the variables of the promising new concepts. These variables are shown in Fig. 11 (middle), and the newly introduced or revised variables are colored in blue. Thus, the initial MOO model is re-formulated for the first time.

6.3. The second Re-OPF iteration (Action C2, D2, E2)

6.3.1. Data generation (Action C2)

The second 300 data sets are generated automatically, based on an updated automation workflow. The workflow is updated by uploading the first re-formulated MOO model.

6.3.2. Information and knowledge extraction (Action D2)

As shown in Fig. 5, one kind of knowledge is extracted during information and knowledge extraction (Action D2), as summarized below.

- Extracting knowledge of promising new concepts (other than

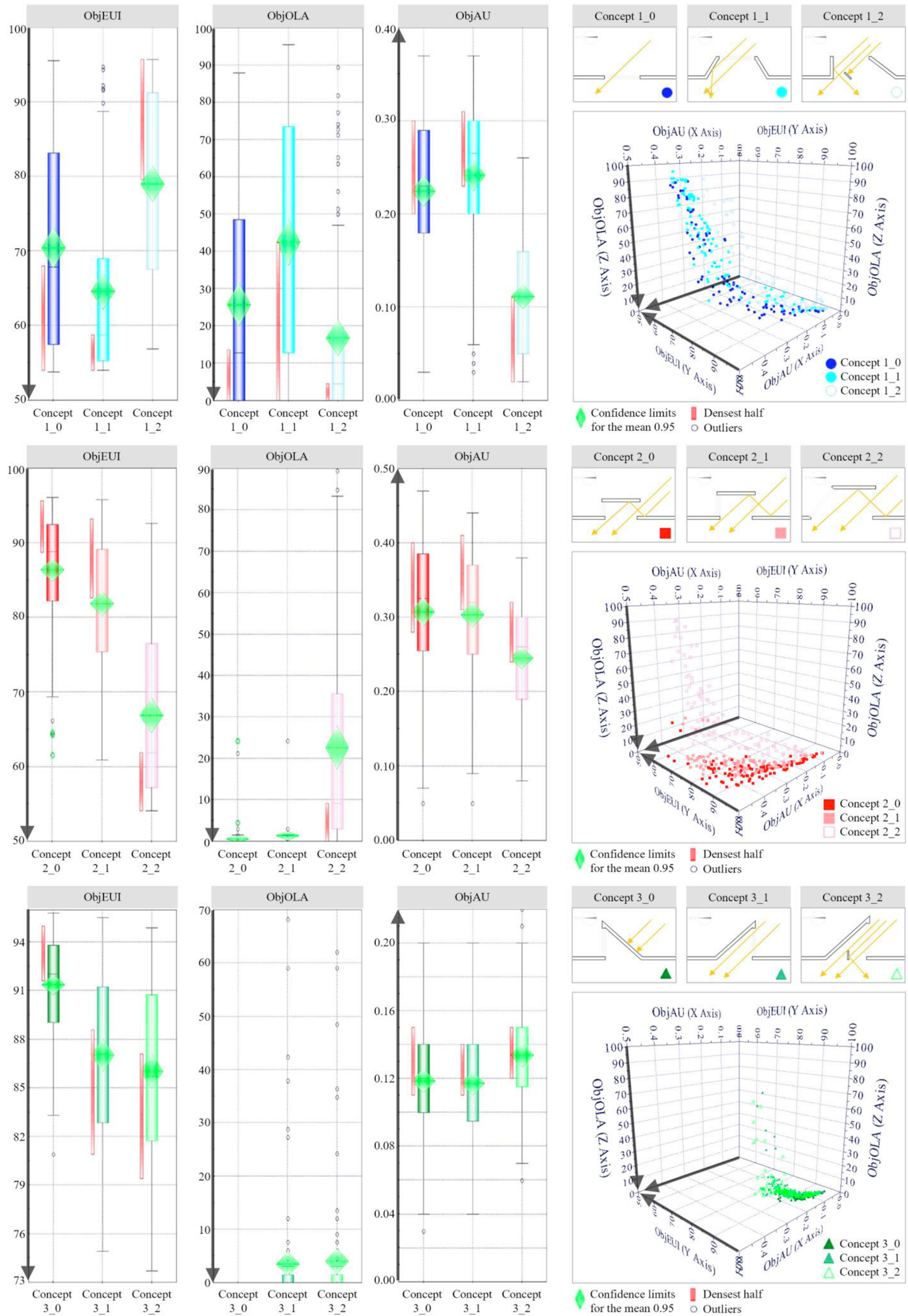


Fig. 15. Box-whisker plots and scatter plots for EUI, OLA, AU of all concepts considered in different phases of the case study. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

previous ones)

Finding quantitatively promising new concepts in the second iteration (Action D2), is similar to that in the first iteration (Action D1). As a result, box-whisker plots and scatter plots for EUI, OLA, AU of the previous concepts (i.e., Concept 1_1, Concept 2_1 and Concept 3_1) are generated (Fig. 15). Via human interpretation, room for possible improvements is found, indicating new design strategies and concepts, as summarized below.

(1) For Concept 1_1, AU is improved as expected, but OLA becomes worse. This indicates that daylight reflected by the inclined north opaque element is over concentrated on certain spots. Thus, a possible solution is to reflect daylight to a wider range. Based on this strategy, Concept 1_2 is produced which makes the north opaque element vertical, adds a shading element (to reflect daylight further into the space), and enlarges the size of the window (to compensate the amount of daylight being blocked out).

(2) For Concept 2_1, EUI is improved as expected. To pursue even better EUI, a more aggressive strategy is to increase daylight from the north. Based on this strategy, Concept 2_2 is produced which expands the opening of the roof bottom surface, and enlarges the size of the north-facing window.

(3) For Concept 3_1, EUI is improved as expected, but AU does not become better. This indicates that the saw-tooth geometry itself has difficulties to evenly spread daylight over the space. Thus, a possible solution is to increase the portion of reflected daylight, while maintaining the total amount of daylight. Based on this strategy, Concept 3_2 is produced which adds a shading element (to increase reflected daylight), expands the opening of the roof bottom surface and enlarges the size of the south-facing window (to compensate the amount of daylight being blocked out).

Judging qualitatively acceptable new concepts in the second iteration (Action D2), is similar to that in the first iteration (Action D1), which needs human subjectivity. Concept 1_2, Concept 2_2 and Concept 3_2 are all judged as aesthetically acceptable. Overall, they are chosen as promising new concepts.

6.3.3. MOO model re-formulation (Action E2)

The above knowledge suggests to conduct: divergent concept generation. Specifically, the parametric geometry model is modified, by using the variables of the promising new concepts. These variables are shown in Fig. 11 (bottom), and the newly introduced or revised variables are colored in red. Thus, the MOO model is re-formulated for the second time.

6.4. The third Re-OPF iteration (Action C3, D3, E3)

6.4.1. Data generation (Action C3)

The third 300 data sets are generated automatically, based on an updated automation workflow. The workflow is updated by uploading the second re-formulated MOO model.

6.4.2. Information and knowledge extraction (Action D3)

As shown in Fig. 5, one kind of knowledge is extracted during information and knowledge extraction (Action D3), as summarized below.

- Extracting knowledge of promising existing concepts (among all explored ones)

Finding quantitatively promising existing concepts in the third iteration (Action D3), is similar to that in the first iteration (Action D1). The quantitative data to be analyzed includes all 900 data sets (not just 300 data sets). As a result, ten clusters of samples are generated (Fig. 16, left). Via human interpretation, three clusters are identified (Fig. 16, right), indicating quantitatively promising existing concepts, as summarized below.

(1) Concept 1_0, Concept 1_1, Concept 2_0, Concept 2_1 and Concept 2_2 are quantitatively promising existing concepts, as the identified clusters of samples belong to these five concepts.
(2) Concept 3_0, Concept 3_1, Concept 3_2 and Concept 1_2 are less quantitatively promising existing concepts, as the identified clusters of samples do not belong to these four concepts.

Judging qualitatively acceptable existing concepts in the third iteration (Action D3), is similar to that in the first iteration (Action D1), which needs human subjectivity. Concept 1_0, Concept 1_1, Concept 2_0, Concept 2_1 and Concept 2_2 are all judged as aesthetically acceptable. Overall, they are chosen as final promising existing concepts.

It is worth noting that promising Concept 3_0 in the first iteration (Action D1) now disappears from the list of final promising concepts in the third iteration (Action D3). This is because Concept 3_0 is overwhelmed by new competitors - promising Concept 1_1, Concept 2_1 and Concept 2_2 derived in the second and third iterations (Action D2 and Action D3). Moreover, it should be clarified that, due to possible inaccuracy of prior knowledge or guesses, concepts generated in later Re-OPF iterations may not necessarily outperform those generated in earlier Re-OPF iterations. For instance, Concept 1_2 performs worse than Concept 1_1 in AU and EUI; Concept 2_2 performs worse than Concept 2_1 in AU and OLA, as shown in Fig. 15.

6.4.3. MOO model re-formulation (Action E3)

The above knowledge suggests to conduct: convergent concept selection. Specifically, the parametric geometry model is modified, by using the variables of the final promising existing concepts. These variables are shown by dashed boxes in Fig. 11. Thus, the MOO model is re-formulated for the third time.

6.5. OPS (Action F, G)

6.5.1. Optimization run (Action F)

Optimization run in this case is for verifying the proposed method, specifically, for verifying benefits of divergent concept generation, and benefits of identified clusters of samples.

First, it is hypothesized that conducting divergent concept generation can help to achieve quantitatively better and qualitatively more diverse Pareto solutions. To verify this hypothesis, Scenario A and B are investigated. In Scenario A, divergent concept generation is not conducted, and optimizations are run for the initial concepts (i.e., Concept 1_0, Concept 2_0, Concept 3_0). In Scenario B, divergent concept generation is conducted, and optimizations are run for the final promising concepts (i.e., Concept 1_0, Concept 1_1, Concept 2_0, Concept 2_1, Concept 2_2). The settings for each optimization include: NSGA-II algorithm [124], an initial population selected from random samples of a concept, a population size of 30, and 10 generations.

Second, it is hypothesized that selecting an initial population from identified clusters of samples can help to achieve better searched solutions and Pareto solutions. To verify this hypothesis, Scenario C and D are investigated. In Scenario C, the initial population is selected from random samples of a concept. In Scenario D, the initial population is selected from identified clusters of samples of a concept. Three of the final promising concepts (i.e., Concept 2_2, Concept 1_0 and Concept 2_1) are selected for optimizations. The optimizations are run for each scenario and each selected concept. The settings for each optimization are the same as the aforementioned ones, except the initial population.

For the verification, it is also necessary to understand an s-Pareto front [125,126] and a hypervolume indicator [127,128]. The s-Pareto front is used to simultaneously consider a set of Pareto fronts of various concepts. It applies Pareto dominance to all optimal solutions on different Pareto fronts, in order to figure out new non-dominated solutions. The hypervolume indicator is used to compare the goodness of

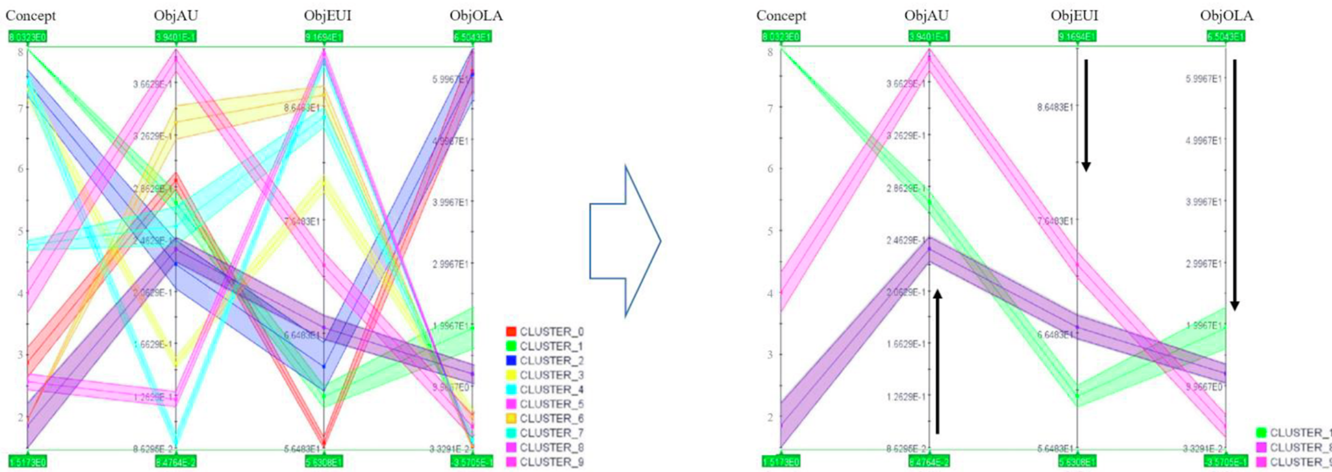


Fig. 16. Clusters in the third Re-OPF iteration. Ten clusters of samples generated (left); three clusters of samples identified (right). CLUSTER_1 consists of samples from Concept 2_2; CLUSTER_8 consists of samples from Concept 1_0 and Concept 1_1; CLUSTER_9 consists of samples from Concept 2_0 and Concept 2_1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

different Pareto fronts. A better Pareto front has a higher hypervolume value. To allow the comparison, the reference point used for calculating the hypervolume is fixed.

6.5.2. Optimization result interpretation (Action G)

Optimization results for verifying the first hypothesis are compared in Fig. 17. In Scenario A, the Pareto fronts of the initial concepts are combined, forming an s-Pareto front which has a lower hypervolume value and consists of solutions from Concept 1_0 and Concept 2_0 (Fig. 17, left). In Scenario B, the Pareto fronts of the final promising concepts are combined, forming an s-Pareto front which has a higher hypervolume value and consists of solutions from Concept 1_0, Concept 2_0, Concept 2_1 and Concept 2_2 (Fig. 17, right). Overall, the s-Pareto front in Scenario B is quantitatively better and qualitatively more diverse, which confirms the first hypothesis.

Optimization results for verifying the second hypothesis are compared in Fig. 18. Two searched solution sets of each selected concept are distributed similarly in the objective space, but with noticeable differences (Fig. 18, top). Further comparison of the performance data shows that the searched solution set of Scenario D is more concentrated and near the desired goal, compared to that of Scenario C. Moreover, two Pareto fronts of each selected concept are also distributed similarly with some differences (Fig. 18, bottom). Further comparison of the hypervolume values shows that the Pareto front of Scenario D is better than that of Scenario C. Overall, the searched solutions and Pareto solutions of Scenario D are better, which confirms the second hypothesis.

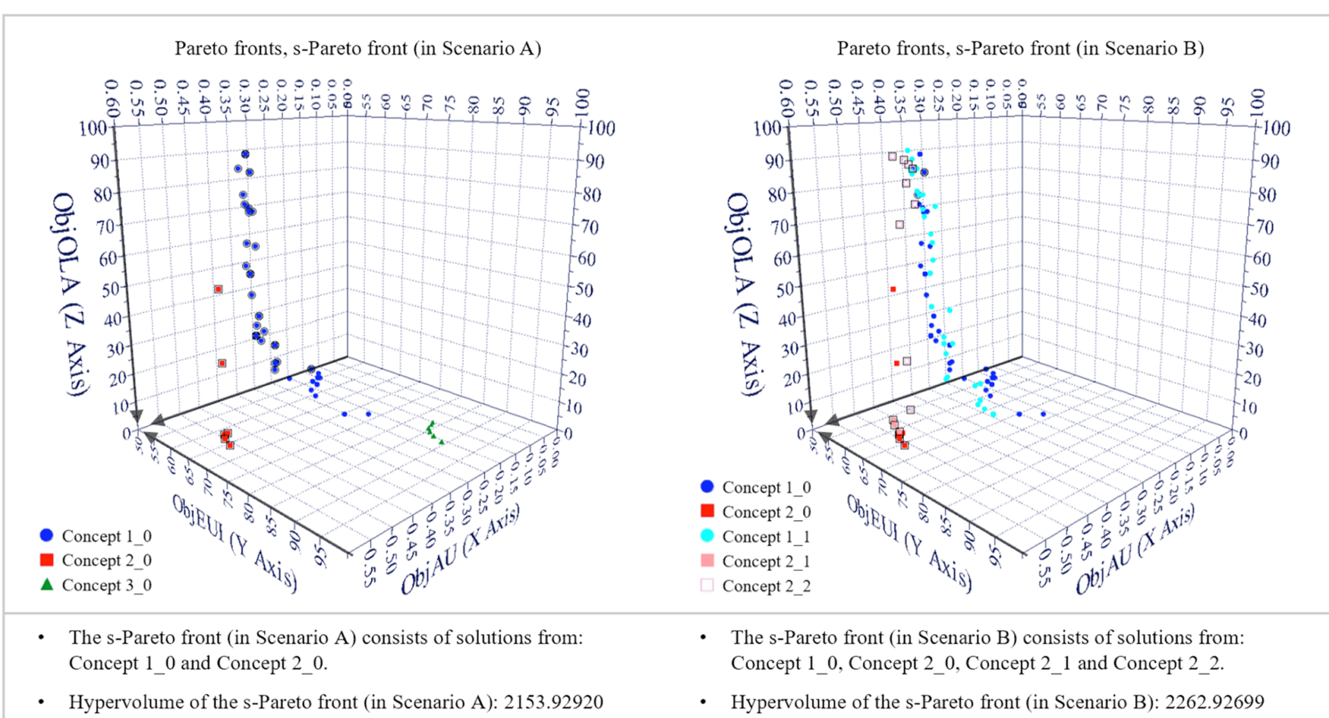


Fig. 17. Optimization result comparison for verifying the first hypothesis. The Pareto fronts and s-Pareto front in Scenario A (left); the Pareto fronts and s-Pareto front in Scenario B (right). The s-Pareto front solutions are marked by black boxes and circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

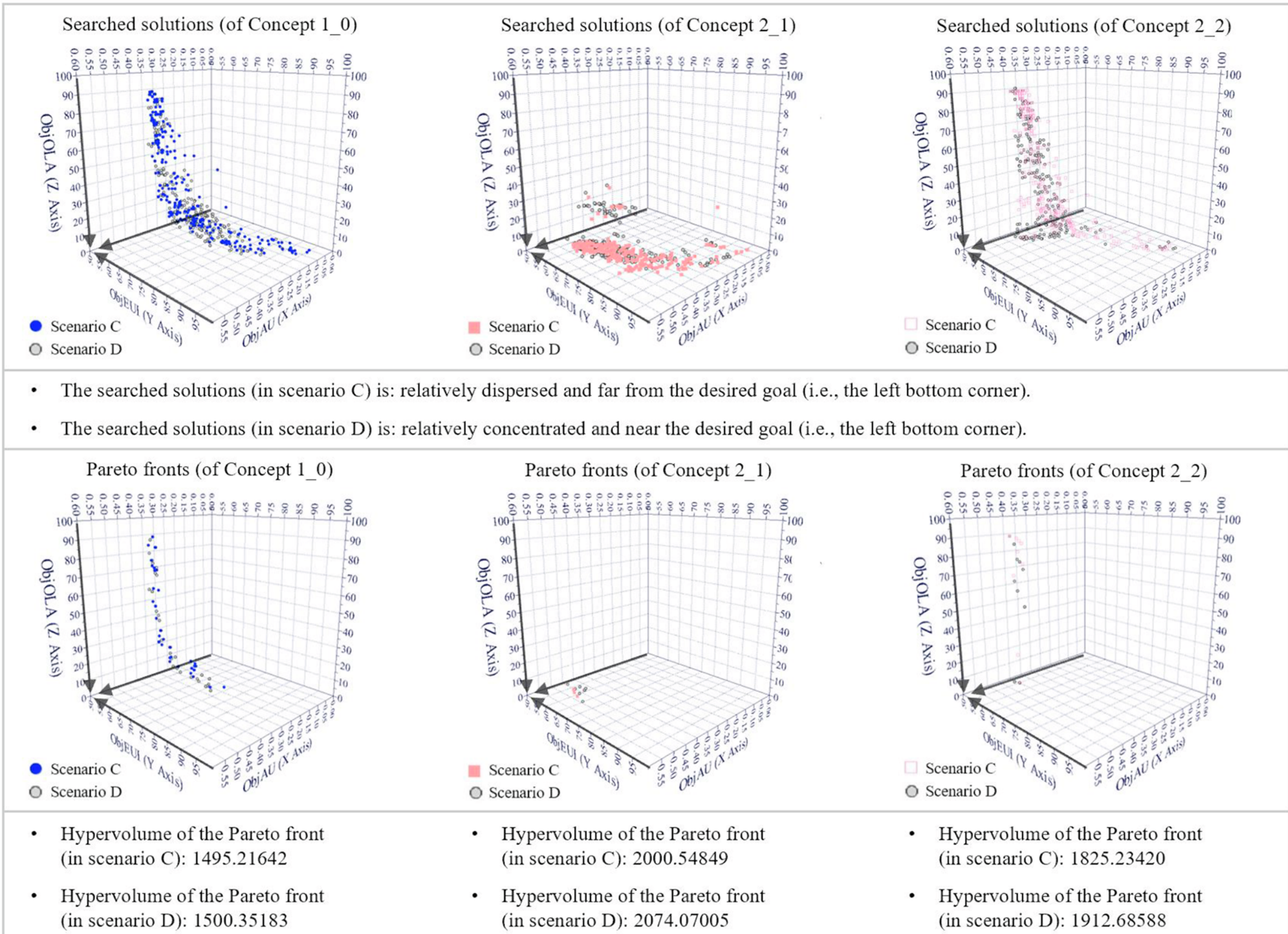


Fig. 18. Optimization result comparison for verifying the second hypothesis. The searched solutions of each selected concept (top); the Pareto fronts of each selected concept (bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7. Conclusion

This section summarizes contributions of this paper and some relevant aspects of the proposed method, and concludes with future research directions and concluding remarks.

7.1. Contributions and discussion

This paper has proposed an innovative SBMOO method for conceptual architectural design exploration. The method is featured with the built-in dynamic and interactive Re-OPF phase. This phase is flexible, allowing designers to include different numbers of Re-OPF iterations, and to perform actions of each iteration in different specific ways. The flexible, reasonably-structured method can lead to greater design success than a rigid, over-structured approach [129]. The method can be supported by a number of computational techniques and a promising integration of software tools - GH-MF integration. With these supports, the method has been applied to a case study of the conceptual design of a top-daylighting system. The results of the case study have confirmed the capability of the proposed method (as described in Section 6.5), and the usability of the adopted techniques and tools.

Human factors may affect the outcome of applying the proposed method. Specifically, designer's prior knowledge is used in quantitative information interpretation; designer's preference is used in qualitative information extraction, and information synthesis. Thus, these human factors can affect what concepts are generated and eventually selected, and hence affect what s-Pareto front is achieved. Nevertheless, as indicated in Fig. 17, the more concepts are explored during Re-OPF iterations, the higher chance a better s-Pareto front is achieved. Since the exploration of multiple initial concepts in multiple Re-OPF iterations requires significant time, designers have to properly allocate available time between the exploration and the consequent optimization.

7.2. Future research directions

This research has several limitations and could be extended in several ways. First, the current case is relatively simplified in term of the completeness of measures and the complexity of geometries, for the convenience of demonstrating the proposed method. In order to show the capability of the method for practical projects, future works can be extended to involve more complete measures and more complex geometries. Second, the current case only conducts performance measure re-formulation in the initial Re-OPF iteration, for the purpose of focusing on concept re-formulation. In order to further emphasize divergent exploration, future works can be extended to conduct performance measure re-formulation in more Re-OPF iterations.

7.3. Concluding remarks

In conclusion, although there are various SBMOO methods for conceptual architectural design, they mostly focus on a one-shot “optimization problem solving”, rather than a continuous “optimization problem framing” or “problem-solution co-evolution” [129] which is a better description of conceptual architectural design. Differently, the proposed SBMOO method highlights the importance of dynamic and interactive Re-OPF, especially the importance of divergent concept generation. The proposed method can help to achieve quantitatively better and qualitatively more diverse Pareto solutions.

Declaration of competing interest

The authors Ding Yang, Michela Turrin, Sevil Sariyildiz, Yimin Sun certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript. The author Danilo di Stefano is an employee of ESTECO SpA, that develops modeFRONTIER, used in the paper as a software tool for design optimization.

Acknowledgement

This research was supported by the Urban System and Environment (USE) Joint Research Centre between South China University of Technology (SCUT) and Delft University of Technology (TU Delft), by the Key Project of National Natural Science Foundation of China (Grant No. 51138004), the 2015 Open Project of State Key Laboratory of Subtropical Building Science, SCUT (Grant No. 2015ZA01), and the TU Delft Sports Engineering Institute Development Fund. We also thank ESTECO for the collaboration based on which the GH-MF integration was previously developed, and for the useful discussions during the case study.

References

[1] K.T. Ulrich, S.D. Eppinger, Product Design and Development, Third edition, McGraw-Hill Irwin, 2004 (ISBN: 0072471468).
[2] K. Meniru, H. Rivard, C. Bédard, Specifications for computer-aided conceptual building design, Des. Stud. 24 (1) (2003) 51–71, https://doi.org/10.1016/S0142-694X(02)00009-1.
[3] M. Turrin, Performance Assessment Strategies, PhD-thesis Delft University of Technology, Department of Architectural Engineering + Technology, 2014, https://doi.org/10.7480/abe.2014.1 (ISBN: 978-94-6186-258-7).
[4] I.S. Sariyildiz, Keynote speech: Performative computational design, ICONARCH-I "Architecture and Technology" International Congress Proceedings Book, Selcuk University and Selcuklu Municipality of Konya, 2012, pp. 313–344 (ISBN: 978-975-448-203-4).
[5] K. Negendahl, Building performance simulation in the early design stage: an introduction to integrated dynamic models, Autom. Constr. 54 (2015) 39–53, https://doi.org/10.1016/j.autcon.2015.03.002.
[6] H.A. Simon, The structure of ill structured problems, Artif. Intell. 4 (3–4) (1973) 181–201, https://doi.org/10.1016/0004-3702(73)90011-8.
[7] D. Navinchandra, Exploration and Innovation in Design: Towards a Computational Model, Springer-Verlag New York, Inc, 1991 (ISBN-13: 978-1-4612-7802-3).
[8] B. Logan, T. Smithers, Creativity and design as exploration, in: J.S. Gero, M.L. Maher (Eds.), Modeling Creativity and Knowledge-based Creative Design, Lawrence Erlbaum Associates, Inc, 1993, pp. 139–173 (ISBN: 0-8058-1153-2).
[9] T. Smithers, D. Corne, P. Ross, On computing exploration and solving design problems, in: J.S. Gero, E. Tyugu (Eds.), Formal Design Methods for CAD, Elsevier Science, Inc, 1994, pp. 293–313 (ISBN: 0-444-81970-3).
[10] J.S. Gero, Towards a model of exploration in computer-aided design, in: J.S. Gero, E. Tyugu (Eds.), Formal Design Methods for CAD, Elsevier Science, Inc, 1994, pp. 315–336 (ISBN: 0-444-81970-3).
[11] M.L. Maher, J. Poon, S. Boulanger, Formalising design exploration as co-evolution: a combined gene approach, in: J.S. Gero, F. Sudweeks (Eds.), Advances in Formal Design Methods for CAD, Springer Science+Business Media, Dordrecht, 1996, pp. 3–30 (ISBN: 978-1-4757-4428-6).
[12] M.L. Maher, J. Poon, Modeling design exploration as co-evolution, Comput. Aided Civ. Infrastruct. Eng. 11 (3) (1996) 195–209, https://doi.org/10.1111/j.1467-8667.1996.tb00323.x.
[13] K. Dorst, N. Cross, Creativity in the design process: co-evolution of problem-solution, Des. Stud. 22 (5) (2001) 425–437, https://doi.org/10.1016/S0142-694X(01)00009-6.
[14] L. Hay, A.H. Duffy, C. McTeague, L.M. Pidgeon, T. Vuletic, M. Grealy, A systematic review of protocol studies on conceptual design cognition: design as search and exploration, Des. Sci. 3 (e10) (2017), https://doi.org/10.1017/dsj.2017.11.
[15] N. Cross, Engineering Design Methods: Strategies for Product Design, Fourth edition, John Wiley & Sons Limited, 2008 (ISBN: 0-471-87250-4).
[16] M. Kanaani, Performativity: The fourth dimension in architectural design, in: M. Kanaani, D. Kopec (Eds.), The Routledge Companion for Architecture Design and Practice, Routledge, 2016, pp. 93–116 (ISBN: 978-1-138-02315-4).
[17] R. Oxman, R. Hammer, S.B. Ari, Performative design in architecture, in: J. Kieferle, K. Ehlers (Eds.), Predicting the Future: Proceedings of the 25th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe 2007), eCAADe, 2007, pp. 227–234 (ISBN: 978-0-9541183-6-5).
[18] R. Oxman, Performance-based design: current practices and research issues, Int. J. Archit. Comput. 6 (1) (2008) 1–17, https://doi.org/10.1260/147807708784640090.
[19] R. Oxman, Emerging paradigms and models in digital design – performance-based architectural design, in: P. Brandon, T. Kocatürk (Eds.), Virtual Futures for Design, Construction & Procurement, Blackwell Publishing Ltd, 2008, pp. 3–20, , https://doi.org/10.1002/9781444302349.ch1 (ISBN: 978-1-4051-7024-6).
[20] R. Oxman, Performative design: a performance-based model of digital architectural design, Environ. Plan. B Urban Anal. City Sci. 36 (6) (2009) 1026–1037, https://doi.org/10.1068/b34149.
[21] B. Kolarevic, Computing the performative in architecture, in: W. Dokonal, U. Hirschberg (Eds.), Digital Design: Proceedings of the 21th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe 2003), eCAADe, 2003, pp. 457–464 (ISBN: 0-9541183-1-6).
[22] B. Kolarevic, Digital morphogenesis, in: B. Kolarevic (Ed.), Architecture in the Digital Age: Design and Manufacturing, Spon Press, 2003, pp. 17–45 (ISBN: 0-415-27820-1).
[23] B. Kolarevic, Back to the future: performative architecture, Int. J. Archit. Comput. 2 (1) (2004) 43–50, https://doi.org/10.1260/1478077041220205.
[24] B. Kolarevic, Computing the performative, in: B. Kolarevic, A. Malkawi (Eds.), Performative Architecture: Beyond Instrumentality, Spon Press, 2005, pp. 193–202 (ISBN: 0-415-70083-3).
[25] B. Kolarevic, Towards the performative in architecture, in: B. Kolarevic, A. Malkawi (Eds.), Performative Architecture: Beyond Instrumentality, Spon Press, 2005, pp. 203–214 (ISBN: 0-415-70083-3).
[26] A.T. Nguyen, S. Reiter, P. Rigo, A review on simulation-based optimization methods applied to building performance analysis, Appl. Energy 113 (2014) 1043–1058, https://doi.org/10.1016/j.apenergy.2013.08.061.
[27] V. Machairas, A. Tsangrassoulis, K. Axarli, Algorithms for optimization of building design: a review, Renew. Sust. Energ. Rev. 31 (2014) 101–112, https://doi.org/10.1016/j.rser.2013.11.036.
[28] D. Meignan, S. Knust, J.M. Frayret, G. Pesant, N. Gaud, A review and taxonomy of interactive optimization methods in operations research, ACM. Trans. Interact. Intell. Syst. 5 (3) (17) (2015), https://doi.org/10.1145/2808234.
[29] J.P. Guilford, The Nature of Human Intelligence, First edition, McGraw-Hill, 1967 (ISBN: 0070251355).
[30] Wikipedia, Kansei engineering, https://en.wikipedia.org/wiki/Kansei_engineering Accessed date: October 28, 2019.
[31] B. Lawson, How Designers Think, Architectural Press, 2005 (ISBN: 0-7506-6077-5).
[32] O. Hoffmann, On modeling human-computer co-creativity, in: S. Kunifuji, G.A. Papadopoulos, A.M.J. Skulimowski, J. Kacprzyk (Eds.), Knowledge, Information and Creativity Support Systems, Springer, 2016, pp. 37–48, , https://doi.org/10.1007/978-3-319-27478-2_3 (ISBN: 978-3-319-27477-5).
[33] D. Yang, S. Ren, M. Turrin, S. Sariyildiz, Y. Sun, Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: a case of conceptual sports building design, Autom. Constr. 92 (2018) 242–269, https://doi.org/10.1016/j.autcon.2018.03.023.
[34] T. Østergård, R.L. Jensen, S.E. Maagaard, Building simulations supporting decision making in early design–a review, Renew. Sust. Energ. Rev. 61 (2016) 187–201, https://doi.org/10.1016/j.rser.2016.03.045.
[35] E. Touloupaki, T. Theodosiou, Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization-a review, Energies 10 (5) (637) (2017), https://doi.org/10.3390/en10050637.
[36] B. Ekici, C. Cubukcuoglu, M. Turrin, I.S. Sariyildiz, Performative computational architecture using swarm and evolutionary optimisation: a review, Build. Environ. 147 (2019) 356–371, https://doi.org/10.1016/j.buildenv.2018.10.023.
[37] D. Newton, Accommodating change and open-ended search in design optimization: dynamic interactive multi-objective evolutionary algorithms for architectural design, in: T. Fukuda, W. Huang, P. Janssen, K. Crolla, S. Alhadidi (Eds.), Learning, Adapting and Prototyping: Proceedings of 23rd International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2018), 2 CAADRIA, 2018, pp. 175–184 (ISBN: 978-988-78917-0-3).
[38] Y.C. Liu, A. Chakrabarti, T. Bligh, Towards an 'ideal' approach for concept generation, Des. Stud. 24 (4) (2003) 341–355, https://doi.org/10.1016/S0142-694X(03)00003-6.
[39] R. Aish, R. Woodbury, Multi-level interaction in parametric design, in: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics: 5th International Symposium (SG 2005), Springer-Verlag Berlin, Heidelberg, 2005, pp. 151–162, , https://doi.org/10.1007/11536482_13 (ISBN: 3-540-28179-5).
[40] J.L. Hensen, R. Lamberts (Eds.), Building Performance Simulation for Design and Operation, Spon Press, 2011(ISBN: 978–0–415–47414–6).
[41] M. Cavazzuti, Optimization Methods: From Theory to Design, Springer-Verlag Berlin, Heidelberg, 2013 (ISBN: 978-3-642-31186-4).
[42] Statistics Solutions, Correlation (Pearson, Kendall, spearman), https://www.statisticssolutions.com/correlation-pearson-kendall-spearman/ Accessed date:

October 28, 2019.
[43] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323, https://doi.org/10.1145/331499.331504.
[44] Rhinoceros, https://www.rhino3d.com/, Accessed date: October 28, 2019.
[45] Grasshopper, http://www.grasshopper3d.com/, Accessed date: October 28, 2019.
[46] modeFRONTIER, http://www.esteco.com/modefrontier, Accessed date: October 28, 2019.
[47] D. Yang, Y. Sun, M. Turrin, P. von Buelow, J. Paul, Multi-objective and multi-disciplinary design optimization of large sports building envelopes: a case study, Proceedings of IASS Annual Symposia, IASS 2015 Amsterdam Symposium: Future Visions - Computational Design, International Association for Shell and Spatial Structures (IASS), 2015, pp. 1–14 (ISSN: 2518-6582).
[48] J.L. Maisel, E. Steinfeld, M. Basnak, K. Smith, M.B. Tauke, Inclusive Design: Implementation and Evaluation, Routledge, 2018 (ISBN: 978-1-138-89034-3).
[49] J. Agte, O. De Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, M. Spieck, MDO: assessment and direction for advancement - an opinion of one international group, Struct. Multidiscip. Optim. 40 (1–6) (2010), https://doi.org/10.1007/s00158-009-0381-5.
[50] T.W. Simpson, J.R.R.A. Martins, Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop, J. Mech. Des. 133 (10) (2011) 101002, , https://doi.org/10.1115/1.4004465.
[51] P. Heiselberg, H. Brohus, A. Hesselholt, H. Rasmussen, E. Seinre, S. Thomas, Application of sensitivity analysis in design of sustainable buildings, Renew. Energy 34 (9) (2009) 2030–2036, https://doi.org/10.1016/j.renene.2009.02.016.
[52] H. Shen, A. Tzempelikos, Sensitivity analysis on daylighting and energy performance of perimeter offices with automated shading, Build. Environ. 59 (2013) 303–314, https://doi.org/10.1016/j.buildenv.2012.08.028.
[53] J.S. Arora, Introduction to Optimum Design, Fourth edition, Academic Press, 2016 (ISBN: 9780128008065).
[54] C. Raquel, X. Yao, Dynamic multi-objective optimization: A survey of the state-of-the-art, in: S. Yang, X. Yao (Eds.), Evolutionary Computation for Dynamic Optimization Problems, Springer, Berlin, Heidelberg, 2013, pp. 85–106, , https://doi.org/10.1007/978-3-642-38416-5_4 (ISBN: 978-3-642-38415-8).
[55] M. Helbig, A.P. Engelbrecht, Dynamic multi-objective optimization using PSO, in: E. Alba, A. Nakib, P. Siarry (Eds.), Metaheuristics for Dynamic Optimization, Springer, Berlin, Heidelberg, 978-3-642-30664-8, 2013, pp. 147–188, , https://doi.org/10.1007/978-3-642-30665-5_8.
[56] R. Azzouz, S. Bechikh, L.B. Said, Dynamic multi-objective optimization using evolutionary algorithms: a survey, in: B. Slim, D. Rituparna, G. Abhishek (Eds.), Recent Advances in Evolutionary Multi-objective Optimization, Springer International Publishing, 2017, pp. 31–70, , https://doi.org/10.1007/978-3-319-42978-6_2 (ISBN: 978-3-319-42978-6).
[57] M. Helbig, A.P. Engelbrecht, Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems, Swarm Evol. Comput. 14 (2014) 31–47, https://doi.org/10.1016/j.swevo.2013.08.004.
[58] W. Trabelsi, R. Azzouz, S. Bechikh, L.B. Said, Leveraging evolutionary algorithms for dynamic multi-objective optimization scheduling of multi-tenant smart home appliances, 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 3533–3540, , https://doi.org/10.1109/CEC.2016.7744237 (ISBN: 978-1-5090-0622-9).
[59] S.K. Curtis, C.A. Mattson, B.J. Hancock, P.K. Lewis, Divergent exploration in design with a dynamic multiobjective optimization formulation, Struct. Multidiscip. Optim. 47 (5) (2013) 645–657, https://doi.org/10.1007/s00158-012-0855-8.
[60] J.M. Cichocka, W.N. Browne, E. Rodriguez, Optimization in the architectural practice - an international survey, in: P. Janssen, P. Loh, A. Raonic, M.A. Schnabel (Eds.), Protocols, Flows and Glitches, Proceedings of the 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2017), CAADRIA, 2017, pp. 387–397.
[61] A.M. Brintrup, J. Ramsden, A. Tiwari, An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization, Comput. Ind. 58 (3) (2007) 279–291, https://doi.org/10.1016/j.compind.2006.06.004.
[62] C.T. Mueller, J.A. Ochsendorf, Combining structural performance and designer preferences in evolutionary design space exploration, Autom. Constr. 52 (2015) 70–82, https://doi.org/10.1016/j.autcon.2015.02.011.
[63] H. Takagi, Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation, Proc. IEEE 89 (9) (2001) 1275–1296, https://doi.org/10.1109/5.949485.
[64] J. Graf, Interactve evolutionary algorithms in design, in: D.W. Pearson, N.C. Steele, R.F. Albrecht (Eds.), Artificial Neural Nets and Genetic Algorithms, Springer, Vienna, 1995, pp. 227–230, , https://doi.org/10.1007/978-3-7091-7535-4_60 (ISBN: 978-3-211-82692-8).
[65] J. Graf, Interactive evolution: a prototyping framework for engineering, Comput. Aided Civ. Infrastruct. Eng. 11 (3) (1996) 185–193, https://doi.org/10.1111/j.1467-8667.1996.tb00322.x.
[66] M. Turrin, P. Von Buelow, R. Stouffs, Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms, Adv. Eng. Inform. 25 (4) (2011) 656–675, https://doi.org/10.1016/j.aei.2011.07.009.
[67] G.J. Barnum, C.A. Mattson, A computationally assisted methodology for preference-guided conceptual design, J. Mech. Des. 132 (12) (2010) 121003, , https://doi.org/10.1115/1.4002838.
[68] P. Janssen, Dexen: a scalable and extensible platform for experimenting with population-based design exploration algorithms, AI EDAM 29 (4) (2015) 443–455, https://doi.org/10.1017/S0890060415000438.
[69] V. Kaushik, P. Janssen, An evolutionary design process: adaptive-iterative explorations in computational embryogenesis, in: R. Stouffs, P. Janssen, S. Roudavski, B. Tunçer (Eds.), Open Systems: Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2013), CAADRIA, 2013, pp. 137–146 (ISBN: 9789881902641).
[70] F.P. Chantrelle, H. Lahmidi, W. Keilholz, M. El Mankibi, P. Michel, Development of a multicriteria tool for optimizing the renovation of buildings, Appl. Energy 88 (4) (2011) 1386–1394, https://doi.org/10.1016/j.apenergy.2010.10.002.
[71] M. Hamdy, A. Hasan, K. Siren, Impact of adaptive thermal comfort criteria on building energy use and cooling equipment size using a multi-objective optimization scheme, Energy Build. 43 (9) (2011) 2055–2067, https://doi.org/10.1016/j.enbuild.2011.04.006.
[72] jEPlus + EA User's Guide, http://www.jeplus.org/wiki/doku.php?id = docs:jeplus_ea:start, Accessed date: October 28, 2019.
[73] S.M. Porritt, P.C. Cropper, L. Shao, C.I. Goodier, Ranking of interventions to reduce dwelling overheating during heat waves, Energy Build. 55 (2012) 16–27, https://doi.org/10.1016/j.enbuild.2012.01.043.
[74] M. Palonen, M. Hamdy, A. Hasan, MOBO a new software for multi-objective building performance optimization, in: E. Wurtz (Ed.), Building Simulation 2013: 13th International Conference of the International Building Performance Simulation Association, IBPSA, 2013, pp. 2567–2574 (ISBN: 978-2-7466-6294-0).
[75] L. Caldas, An Evolution-based Generative Design System: Using Adaptation to Shape Architectural Form, PhD-thesis, Massachusetts Institute of Technology, Department of Architecture, 2001, http://hdl.handle.net/1721.1/8188, Accessed date: 28 October 2019.
[76] L. Caldas, GENE_ARCH: an evolution-based generative design system for sustainable architecture, in: I.F.C. Smith (Ed.), Intelligent Computing in Engineering and Architecture, Springer, 2006, pp. 109–118, , https://doi.org/10.1007/11888598_12 (ISBN: 978-3-540-46246-0).
[77] L. Caldas, Generation of energy-efficient architecture solutions applying GENE_ARCH: an evolution-based generative design system, Adv. Eng. Inform. 22 (1) (2008) 59–70, https://doi.org/10.1016/j.aei.2007.08.012.
[78] L.G. Caldas, L.K. Norford, A design optimization tool based on a genetic algorithm, Autom. Constr. 11 (2) (2002) 173–184, https://doi.org/10.1016/S0926-5805(00)00096-0.
[79] L.G. Caldas, L.K. Norford, Shape generation using pareto genetic algorithms: integrating conflicting design objectives in low-energy architecture, Int. J. Archit. Comput. 1 (4) (2003) 503–515, https://doi.org/10.1260/147807703773633509.
[80] J.A. Wright, A. Brownlee, M.M. Mourshed, M. Wang, Multi-objective optimization of cellular fenestration by an evolutionary algorithm, J. Build. Perform. Simul. 7 (1) (2014) 33–51, https://doi.org/10.1080/19401493.2012.762808.
[81] K. Shea, A. Sedgwick, G. Antonuntto, Multicriteria optimization of paneled building envelopes using ant colony optimization, in: I.F.C. Smith (Ed.), Intelligent Computing in Engineering and Architecture, Springer, 2006, pp. 627–636, , https://doi.org/10.1007/11888598_56 (ISBN: 978-3-540-46246-0).
[82] Z.X. Conti, Multi-Objective Optimisation of Building Geometry for Energy Consumption and View Quality, Master-thesis, University of Bath, Department of Architecture and Civil Engineering, 2013, https://researchportal.bath.ac.uk/en/studentTheses/multi-objective-optimisation-of-building-geometry-for-energy-cons, Accessed date: October 28, 2019.
[83] Z.X. Conti, P. Shepherd, P. Richens, Multi-objective optimisation of building geometry for energy consumption and view quality, in: B. Martens, G. Wurzer, T. Grasl, W.E. Lorenz, R. Schaffranek (Eds.), Real Time: Proceedings of the 33rd Conference on Education in Computer Aided Architectural Design in Europe (eCAADe 2015), eCAADe, 2015, pp. 287–294 (ISBN: 9789491207082).
[84] J.M Gagne, M. Andersen, Multi-objective facade optimization for daylighting design using a genetic algorithm, Proceedings of SimBuild 4 (1) (2010) 110–117, https://ibpsa-usa.org/index.php/ibpusa/article/view/278, Accessed date: 28 October 2019.
[85] D.J. Gerber, S.H.E. Lin, Designing-in performance through parameterisation, automation, and evolutionary algorithms: 'HDS BEAGLE 1.0', in: T. Fischer, K. De Biswas, J.J. Ham, R. Naka, W. Huang (Eds.), Beyond Codes and Pixels: Proceedings of the 17th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2012), CAADRIA, 978-988-19026-3-4, 2012, pp. 141–150.
[86] D.J. Gerber, S.H.E. Lin, Designing in complexity: simulation, integration, and multidisciplinary design optimization for architecture, Simulation 90 (8) (2014) 936–959, https://doi.org/10.1177/0037549713482027.
[87] DesignBuilder, https://designbuilder.co.uk/, Accessed date: October 28, 2019.
[88] P. Janssen, C. Basol, K.W. Chen, Evolutionary developmental design for non-programmers, in: T.S. Zupancic, M. Juvancic, S. Verovsek, A. Jutraz (Eds.), Respecting Fragile Places: Proceedings of the 29th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe 2011), eCAADe, 2011, pp. 245–252 (ISBN: 978-9-4912070-1-3).
[89] P. Janssen, Evo-Devo in the sky, in: R. Stouffs, S. Sariyildiz (Eds.), Computation and Performance: Proceedings of the 31st Conference on Education in Computer Aided Architectural Design in Europe (eCAADe 2013), eCAADe, 2013, pp. 205–214 (ISBN: 978-94-91207-04-4).
[90] P. Von Buelow, ParaGen: performative exploration of generative systems, Journal of the International Association for Shell and Spatial Structures 53 (4) (2012) 271–284, https://www.semanticscholar.org/paper/PARAGEN-%3A-PERFORMATIVE-EXPLORATION-OF-GENERATIVE-Buelow/3d62bbfe876a57106deaf200012e5db2b837cadc#citing-papers, Accessed date: October 28, 2019.
[91] P. Von Buelow, Genetically enhanced parametric design in the exploration of architectural solutions, in: P.J.S. Cruz (Ed.), Structures and Architecture - Beyond Their Limits, CRC Press, 2016, pp. 675–683 (ISBN: 978-1-138-02651-3).
[92] R. Vierlinger, K. Bollinger, Accommodating change in parametric design, in:

D.J. Gerber, J. Sanchez, A. Huang (Eds.), ACADIA 2014 Design Agency: Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA, 2014, pp. 609–618 (ISBN: 192672447X).
[93] K. Negendahl, T.R. Nielsen, Building energy optimization in the early design stages: a simplified method, Energy Build. 105 (2015) 88–99, https://doi.org/10.1016/j.enbuild.2015.06.087.
[94] N.C. Brown, C.T. Mueller, Design for structural and energy performance of long span buildings using geometric multi-objective optimization, Energy Build. 127 (2016) 748–761, https://doi.org/10.1016/j.enbuild.2016.05.090.
[95] N. Brown, J.I.F. de Oliveira, J. Ochsendorf, C. Mueller, Early-stage integration of architectural and structural performance in a parametric multi-objective design tool, in: P.J.S. Cruz (Ed.), Structures and Architecture - Beyond Their Limits, CRC Press, 2016, pp. 1103–1111.
[96] Stormcloud, http://digitalstructures.mit.edu/page/tools#stormcloud-interactive-evolutionary-exploration-for-grasshopper, Accessed date: October 28, 2019.
[97] R.A. Danhaive, C.T. Mueller, Combining parametric modeling and interactive optimization for high-performance and creative structural design, Proceedings of IASS Annual Symposia, IASS 2015 Amsterdam Symposium: Future Visions - Computational Design, International Association for Shell and Spatial Structures (IASS), 2015, pp. 1–11 (ISSN: 2518-6582).
[98] F. Flager, B. Welle, P. Bansal, G. Soremekun, J. Haymaker, Multidisciplinary process integration and design optimization of a classroom building, Journal of Information Technology in Construction 14 (2009) 595–612 http://www.itcon.org/2009/38.
[99] D. Davis, M. Burry, J. Burry, The flexibility of logic programming: parametrically regenerating the Sagrada Família, in: C. Herr, N. Gu, S. Roudavski, M.A. Schnabel (Eds.), Circuit Bending, Breaking and Mending: Proceedings of the 16th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2011), CAADRIA, 2011, pp. 29–38 (ISBN: 978-988-19026-2-7).
[100] M.D. McKay, R.J. Beckman, W.J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (2) (1979) 239–245, https://doi.org/10.2307/1268522.
[101] M. Bernal, J.R. Haymaker, C. Eastman, On the role of computational support for designers in action, Des. Stud. 41 (Part B) (2015) 163–182, https://doi.org/10.1016/j.destud.2015.08.001.
[102] ESTECO SpA, modeFRONTIER User Guide, https://www.esteco.com/modefrontier, Accessed date: October 28, 2019.
[103] A. Clarich, R. Russo, Use of multi-variate-data-analysis, fast algorithms and grid automation in modeFrontier for multi-objective optimization, in: I.H. Tuncer (Ed.), Proceedings of the ECCOMAS Thematic Conference on CFD & Optimization, ECCOMAS, 2011(ISBN: 978-605-61427-4-1).
[104] H. Díaz, L.F. Alarcón, C. Mourgues, S. García, Multidisciplinary design optimization through process integration in the AEC industry: strategies and challenges, Autom. Constr. 73 (2017) 102–119, https://doi.org/10.1016/j.autcon.2016.09.007.
[105] T. Kohonen, Self-Organizing Maps, Third edition, Springer-Verlag, Berlin Heidelberg, 2001, https://doi.org/10.1007/978-3-642-56927-2 (ISBN: 978-3-540-67921-9).
[106] D. Di Stefano, Multivariate analysis algorithms in modeFRONTIER v4, ESTECO technical report 2009-001, https://www.esteco.com/modefrontier, Accessed date: October 28, 2019.
[107] J. Himberg, J. Ahola, E. Alhoniemi, J. Vesanto, O. Simula, The self-organizing map as a tool in knowledge engineering, in: N.R. Pal (Ed.), Pattern Recognition in Soft Computing Paradigm, World Scientific, 978-981-02-4491-0, 2001, pp. 38–65, , https://doi.org/10.1142/9789812811691_0002.
[108] J. Vesanto, J. Ahola, Hunting for correlations in data using the self-organizing map, in: H. Bother, E. Oja, E. Massad, C. Haefke (Eds.), Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA 99), ICSC Academic Press, 1999, pp. 279–285 (ISBN: 3906454185).
[109] J. Vesanto, SOM-based data visualization methods, Intell. Data Anal. 3 (2) (1999) 111–126, https://doi.org/10.1016/S1088-467X(99)00013-X.
[110] A. Köhler, M. Ohrnberger, F. Scherbaum, Unsupervised pattern recognition in continuous seismic wavefield records using self-organizing maps, Geophys. J. Int. 182 (3) (2010) 1619–1630, https://doi.org/10.1111/j.1365-246X.2010.04709.x.
[111] M. Yoshimura, K. Izui, Smart optimization of machine systems using hierarchical genotype representations, J. Mech. Des. 124 (3) (2002) 375–384, https://doi.org/10.1115/1.1486013.
[112] Y.K. Wong, J. Sharp, A specification and design methodology based on data flow principles, in: J. Sharp (Ed.), Data Flow Computing: Theory and Practice, Ablex Publishing Corp, 1992, pp. 37–79 (ISBN: 978-0-89391-654-1).
[113] D. Davis, J. Burry, M. Burry, Untangling parametric schemata: Enhancing collaboration through modular programming, in: P. Leclercq, A. Heylighen, G. Martin (Eds.), Designing Together: Proceedings of the 14th International Conference on Computer Aided Architectural Design (CAAD Futures 2011), University of Liege, 2011, pp. 55–68.
[114] Ladybug Tools, https://www.ladybug.tools/, Accessed date: October 28, 2019.
[115] Daysim, https://daysim.ning.com/, Accessed date: October 28, 2019.
[116] EnergyPlus, https://energyplus.net/, Accessed date: October 28, 2019.
[117] GenerativeComponents, https://www.bentley.com/en/products/product-line/modeling-and-visualization-software/generativecomponents, Accessed date: October 28, 2019.
[118] Dynamo Studio, https://www.autodesk.com/products/dynamo-studio/overview, Accessed date: October 28, 2019.
[119] Digital Project, https://www.digitalproject3d.com/, Accessed date: October 28, 2019.
[120] Houdini, https://www.sidefx.com/, Accessed date: October 28, 2019.
[121] ModelCenter, https://www.phoenix-int.com/, Accessed date: October 28,2019.
[122] R. Sileryte, A. D'Aquilio, D. Di Stefano, D. Yang, M. Turrin, Supporting exploration of design alternatives using multivariate analysis algorithms, in: R. Attar, A. Chronis, S. Hanna, M. Turrin (Eds.), 2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design, Simulation Councils Inc, 2016, pp. 215–222 (ISBN: 978-1-365-05872-1).
[123] D. Yang, Y. Sun, D. Di Stefano, M. Turrin, S. Sariyildiz, Impacts of problem scale and sampling strategy on surrogate model accuracy: an application of surrogate-based optimization in building design, 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 4199–4207, , https://doi.org/10.1109/CEC.2016.7744323 (ISBN: 978-1-5090-0622-9).
[124] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, https://doi.org/10.1109/4235.996017.
[125] C.A. Mattson, A. Messac, Concept selection using s-Pareto frontiers, AIAA J. 41 (6) (2003) 1190–1198, https://doi.org/10.2514/2.2063.
[126] C.A. Mattson, A. Messac, Pareto frontier based concept selection under uncertainty, with visualization, Optim. Eng. 6 (1) (2005) 85–115, https://doi.org/10.1023/B:OPTE.0000048538.35456.45.
[127] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms - a comparative case study, in: A.E. Eiben, T. Bäck, M. Schoenauer, H.P. Schwefel (Eds.), Parallel Problem Solving from Nature — PPSN V, Springer-Verlag, Berlin Heidelberg, 1998, pp. 292–301, , https://doi.org/10.1007/BFb0056872 (ISBN: 978-3-540-65078-2).
[128] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: on the design of Pareto-compliant indicators via weighted integration, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), Evolutionary Multi-Criterion Optimization, Springer-Verlag, Berlin Heidelberg, 2007, pp. 862–876, , https://doi.org/10.1007/978-3-540-70928-2_64 (ISBN: 978-3-540-70927-5).
[129] N. Cross, Design cognition: Results from protocol and other empirical studies of design activity, in: C. Eastman, M. McCracken, W. Newstetter (Eds.), Design Knowing and Learning: Cognition in Design Education, First edition, Elsevier Science, 2001, pp. 79–103 (ISBN: 978-0-08-043868-9).