Design as Exploration

Multi-Objective and Multi- Disciplinary Optimization (MOMDO) of Indoor Sports Halls

Yang, D.

Publication date
2022

Document Version
Final published version

Citation (APA)
Yang, D. (2022). *Design as Exploration: Multi-Objective and Multi- Disciplinary Optimization (MOMDO) of Indoor Sports Halls*. [Dissertation (TU Delft), Delft University of Technology].

Design as Exploration

Multi-Objective and Multi-Disciplinary Optimization (MOMDO) of Indoor Sports Halls

Ding Yang

Design as Exploration

Multi-Objective and Multi-Disciplinary Optimization (MOMDO) of Indoor Sports Halls

Ding Yang

TOC

A+BE | Architecture and the Built Environment | TU Delft BK

22#25

Design | Sirene Ontwerpers, Véro Crickx

Cover photo | Courtesy of Sun Yimin Studio

ISBN 978-94-6366-643-5
ISSN 2212-3202

This dissertation is open access at https://doi.org/10.7480/abe.2022.25

Design as Exploration

Multi-Objective and Multi-Disciplinary Optimization (MOMDO) of Indoor Sports Halls

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof.dr.ir. T.H.J.J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on
Friday 23 December 2022 at 10.00 o'clock

by

Ding YANG
Bachelor of Architecture, South China University of Technology, P.R. China
born in Liuzhou, Guangxi, P.R. China

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr.ir.arch. I.S. Sariyildiz	Delft University of Technology, promotor
Prof.dr.ir. Y. Sun	South China University of Technology, promotor
Dr. M. Turrin	Delft University of Technology, copromotor

Independent members:

Prof.dr. C. Poloni	University of Trieste
Prof.dr. P.W.C. Chan	Delft University of Technology
Prof.dr. L.C.M. Itard	Delft University of Technology
Prof.dr.ir. P.J.M. van Oosterom	Delft University of Technology
Prof.ir. P.G. Luscuere	Delft University of Technology, reserve member

This research was funded by the China Scholarship Council (CSC).

The formulation of a problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill.

Albert Einstein

Preface

This research was started based on my learning and working experiences at Sun Yimin Studio, Architectural Design and Research Institute of South China University of Technology (SCUT), in the period 2009-2013. During that period, I began to develop my interest in computational design methods and apply them in the practical design of a number of large public building projects. Meanwhile, based on the design practice, I started to reflect on the important role of building performances and the proper use of relevant computational means in achieving high-performing design solutions. This reflection brought me to the field of performance-based design, with a particular focus on its use in the conceptual design phase of sports buildings.

Since 2014, I have had the opportunity to further my research in the Chair of Design Informatics led by Prof. Sevil Sariyildiz, Department of Architectural Engineering and Technology, Delft University of Technology (TU Delft). Benefiting from the strengths of the chair in computational design research and the collaboration with Arup Amsterdam and ESTECO SpA, I was able to focus on optimization and other relevant computational techniques, aiming at developing a Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method suitable for use in ill-defined conceptual architectural design.

This research was based on the Agreement for Joint Supervision and Double Degree of Doctoral Research between TU Delft and SCUT, and supported by Arup Amsterdam, ESTECO SpA, and the Urban Systems and Environment - Joint Research Centre between TU Delft and SCUT. This research was funded by: China Scholarship Council, South China University of Technology, The National Natural Science Foundation of China, State Key Laboratory of Subtropical Building Science, TU Delft Sports Engineering Institute, and Arup Amsterdam.

Acknowledgements

The completion of this research is a milestone in my academic career. Hopefully, the research outcomes will be inspirational and helpful for people who want to achieve high-performing design solutions. To me, the journey of conducting this research has been precious and unforgettable, although it had ups and downs. Reaching the end of my Ph.D. journey, I have so many people to thank for their generous help and support in various ways over the years. This research would have not been possible without their guidance, contribution, and encouragement.

First, I would like to express my deepest gratitude to all my supervisors, Prof. Sevil Sariyildiz, Prof. Yimin Sun, and Dr. Michela Turrin. It has been my privilege to be supervised by them. They have provided me with a terrific research environment and contributed tremendously to the completion of my research.

I am deeply grateful to my promotor Prof. Sevil Sariyildiz, the Chair of Design Informatics, for welcoming me as a member of her research group. Her profound guidance and advice in performative computational design have been essential for my research. She has always been very patient and supportive in her supervision, encouraging me to develop my own ideas with confidence, helping me to clarify my thoughts, and allowing me to reach my full potential. Moreover, she has also been very kind and caring to me and my family, which always makes me feel warm inside.

I am sincerely thankful to my promotor Prof. Yimin Sun for giving me the opportunity to learn and practice in his architectural office - Sun Yimin Studio. With his great support, I have been able to deeply engage in a number of design and research projects in China, some of which later received important awards from the government and industry. I have learned a lot from his extraordinary expertise and professional experience, especially in terms of sports building design. This allows me to develop myself as a practical architect and a Ph.D. researcher, which is also crucial for conducting this research.

I truly appreciate my copromotor Dr. Michela Turrin for her excellent daily supervision and endless support regardless of time and place. She has been the person who has spent numerous meetings with me identifying critical issues, brainstorming new ideas, discussing technical details, and examining the results of my work. Thanks to her open mind and rigorous thinking, her guidance has

always been constructive. She has also encouraged me to collaborate with research partners and to present and discuss my work at several academic events, which has made a big difference in my research. In addition, her positive life attitude and wonderful personality have been an inspiration to me.

The joint supervision has been based on the agreement between Delft University of Technology (TU Delft) and South China University of Technology (SCUT). I appreciate both universities for making it possible; and also appreciate the China Scholarship Council and SCUT for providing me with the scholarships.

I would like to thank other members of the doctoral committee, Prof.dr. C. Poloni, Prof.dr. P.W.C. Chan, Prof.dr. L.C.M. Itard, Prof.dr.ir. P.J.M. van Oosterom and Prof.ir. P.G. Luscuere, for their time to review my dissertation.

This research has benefited from several research partners. My sincere gratitude also goes to them, including structural engineers from Arup Amsterdam, the Netherlands; software developers from ESTECO SpA, Italy; experts from the University of Michigan, United States; and researchers in the Chair of Design Informatics at TU Delft.

I would like to thank Prof. Joop Paul for allowing me to collaborate with Arup Amsterdam. The collaboration on the project Multi-objective Multi-disciplinary Optimization of Structural and Building Envelope Design has contributed to the first case study of my dissertation. I appreciate very much Mr. Shibo Ren, the senior structural engineer at Arup Amsterdam, for helping me to develop the structural model of the case and edit relevant parts of our joint paper; I have been amazed by his wonderful work at the intersection between structural engineering and architecture. Moreover, I would like to express my heartfelt thanks once again to Prof. Joop Paul for his generous guidance and help during the early phase of my research.

I would like to thank Prof. Carlo Poloni for allowing me to collaborate with ESTECO SpA. The collaboration on the development of the Grasshopper-modeFRONTIER (Gh-mF) node has contributed to the software workflow development and case studies of my dissertation. I am very grateful to Mr. Fabio Delli Carpini, the IT consultant at ESTECO SpA, for helping me to realize the seamless integration between Grasshopper and modeFRONTIER. I believe this integration will have a bright future in architectural design optimization. Moreover, I would like to express my sincere thanks to Mr. Danilo Di Stefano, the product manager of ESTECO SpA, for his input in our joint paper; and the same sincere thanks go to Dr. Alberto Clarich, the head of the engineering team of ESTECO SpA, and Mr. Zhongli Wen, the application engineer at ESTECO SpA, for their generous support during the use of modeFRONTIER

I would like to thank Prof. Peter von Buelow from the University of Michigan for supporting my research. He has been so kind to allow me to use his amazing tool ParaGen, help me to test the model of the first case study with his tool, share with me his vast knowledge and experience in optimization, and dedicate his time to review my papers (even though he was not listed in the authorship, and was in his most hectic and difficult time). I appreciate very much all the generous support from the bottom of my heart. While working with him, I have been deeply impressed by his work passion, open mind, and productivity. He has been a wonderful role model to me as a researcher.

I would like to thank Ms. Rusne Sileryte and Mr. Antonio D'Aquilio for being my research partners. We have collaborated on the project 'Multi-objective Optimization of Sports Building Envelope', which has contributed to the software workflow development of my dissertation. I am very thankful to Ms. Rusne Sileryte for helping me to realize some parts of the workflow and thank Mr. Antonio D'Aquilio for his advice in energy simulation. I thank them also for their amazing contribution to our joint papers, one of which has received the Best Paper Award. They have been friendly, supportive, intelligent, hardworking, and perfect collaborators. It has been a very enjoyable experience working with and learning from them.

In addition to my research partners, I would like to thank other experts and researchers who have also provided me with generous help in various ways. From the SCUT side, I thank Prof. Shuoxian Wu, Prof. Chunyang Zhang, Prof. Yiqiang Xiao, Prof. Yufeng Zhang, Prof. Changxin Peng, Prof. Guoguang Wang, Prof. Jing Wang, Prof. Ping Su, Dr. Lu Wang, and Dr. Minzhi Li for providing their inspiring comments and invaluable suggestions at different phases of my research; I thank Dr. Wenbo Sun for sharing his expertise in structural design, and I also thank Dr. Tianxiang Leng and Dr. Lu Xiong for bringing me into the world of computational design. From the TU Delft side, I thank Prof. Özer Ciftcigolu and Dr. Michael Bittermann for letting me know more about optimization and computational cognition, and providing critical comments and advice on my research; I thank Dr. Peter van den Engel, Dr. Xiaoyu Du, Dr. Martin Tenpierik, and Mr. Michael Fremouw for sharing their expertise in climate design, I thank Dr. Pirouz Nourian for sharing his expertise in parametric design; I thank Dr. Jeroen Coenders for sharing his expertise in structural design and optimization; I thank Dr. Haiqiang Wang and Dr. Ali Elham for letting me know more about the application of multi-disciplinary optimization in the aerospace industry, and brainstorming ideas for my research from a cross-disciplinary perspective; I thank Mr. Paul de Ruiter and Mr. Henry Kiksen for always providing me IT support for my research, and I also thank all secretaries of the Architectural Engineering + Technology department for their very kind and prompt support. Moreover, I appreciate Dr. Rudi Stouffs, Dr. Patrick Janssen, and Dr. Ji Zhang for sharing their research and experience at the National University of

Singapore; and thank Prof. Vincent Nadin and Dr. Lei Qu for providing me the chance to join the events at the Urban Systems and Environment Research Centre between TU Delft and SCUT, which have broadened my horizons.

During my Ph.D. journey, I have been so fortunate to get to know so many excellent people from different backgrounds, to learn from them, and to grow friendships with them. They have helped me a lot on many occasions and in various ways. My sincere gratitude also goes to them. Meantime, I apologize for not being able to mention all their names due to the limit of pages.

I would like to express my sincere thanks to Ms. Linghua Li and her family. She contributed a lot to reaching the Agreement for Joint Supervision and Double Degree of Doctoral Research between TU Delft and SCUT, and helped me to establish contact with the research team at TU Delft. Without her help, my research in TU Delft would not have been possible. She and her family have always been very kind and caring and helped me a lot in every aspect of my life in the Netherlands. I would definitely cherish our friendship and all the moments of relaxation we spent together. Moreover, my sincere thanks also go to Ms. Bo Song who has been the management assistant of the Architectural Engineering + Technology department. She has a great personality and has always been very supportive and warm-hearted. With her great help, I was able to quickly adapt to the new environment when I first arrived at the department, and I have been introduced to many excellent professionals across different fields.

I would like to thank my colleagues and friends at TU Delft. Wang Pan, Dejian Peng, Xiao Guo, Wei Dai, Weibin Lin, Peng Gao, and Bowei Huang have been more than friends to me. They have been my companions and support in all the events of my Dutch life. They have been good listeners when I needed an ear, and great encouragers when I got stuck. I cherish the days we helped and supported each other over these years. Yu-chou Chiang has been a close friend of mine since we first met in Delft. We have talked about everything at work and in life. I have enjoyed sharing perspectives and having frank discussions and constructive debates with him on various topics. I appreciate him for all those wonderful talks which were always enlightening. Miktha Farid Alkadri has been my great desk mate in the Ph.D. room. We spent a lot of time brainstorming and discussing together. He has set a model for me of hard work, concentration, and persistence. Berk Ekici and Cemre Çubukcuoglu have been my friends since we first met in Japan for a conference. I have been amazed by their great works and have got a lot of inspiration from them. Moreover, I would like to thank Yuting Tai, Jiaxiu Cai, Meng Meng, Mei Liu, Yunsong Han, Anxiao Zhang, Tiantian Du, Dadi Zhang, Nan Bai, Zhejun Pei, Xiaoyu Yuan, and all, for their companionship over these years. The days with them have brought me a lot of joy. I will never forget those moments we spent together in every corner of Delft.

I would like to thank my colleagues and friends in SCUT. I thank Fenqiang Wang, Lian Tao, Yeqing Huang, Weikang Ye, Yonggang Shen, Sheng Xia, Fan Peng, Ying Xu, Fang Deng, Yanyan Liang, Wenyu Zhang, Zhujian Huang, Haiquan Li, and all, for their kind help and supports regardless of distances.

Last but foremost, I dedicate all my achievements to my beloved family. My lovely wife and daughter, my dear parents, and all my family members are the true “silent heroes” of my life. Words are powerless to express my endless gratitude and love for them. They make my life worth living.

Contents

List of Tables

List of Figures

Summary

Design is goal-directed activity. Its goals are often expressed in the form of "*performance*" which represents how well or badly a design works. Thus, Performance-Based Design (PBD) approaches are a natural choice to improve the quality of a design. When focusing on the field of building design, some well-known architectural firms have adopted Performance-Based Building Design (PBBD) approaches which show relevant benefits, though these approaches are not commonly used by the entire architectural community. Great enhancements to these approaches include optimal-design paradigms (i.e., the paradigms where the design of a system is formulated or partially formulated as a problem of optimization). Recently, these enhanced approaches have made their way to conceptual architectural design and allowed a growing number of designers to benefit from them. In particular, the approaches enhanced by Multi-Disciplinary Optimization (MDO) and Multi-Objective Optimization (MOO) paradigms are receiving increasing attention, given their ability to deal with multiple performance criteria from different design disciplines during the conceptual design of complex buildings. Nevertheless, when looking at the current optimal-design paradigms, there is often a lack of a way to ensure the achievement of a reliable optimization problem, which hinders reliable design solutions despite the use of advanced optimization algorithms. This is the main problem addressed in this research. Achieving such a way is especially important for conceptual architectural design because in this phase the design tasks are generally ill-defined and the optimization problems formulated are usually ill-structured. It can help designers achieve more reliable conceptual solutions and make more informed early decisions so that they can benefit more from the enhanced Performance-Based Building Design (PBBD) approaches.

A promising way to address the above problem is to highlight the importance of Optimization Problem Re-Formulation (Re-OPF). For conceptual architectural design, the formulation of an optimization problem is actually more essential than its solution. That is to say, an ideal optimal-design paradigm should shift the priority from Optimization Problem Solving (OPS) to Optimization Problem Formulation (OPF), more specifically to dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). Such re-formulation allows designers to add and/or remove objectives, constraints, and design variables (i.e., shift objective space and design space to include unexplored areas and/or exclude existing areas). Thus, the re-formulation can be seen as a useful way to achieve a more reliable optimization

problem. Further, to ensure more informed re-formulation, it is crucial to extract relevant knowledge. The knowledge includes, but is not limited to, which objectives and constraints should be considered more meaningful, which design variables should be considered more promising, and how these elements interact with each other. The knowledge can be useful not only for convergent re-formulation but also for divergent re-formulation. It can be derived by analyzing quantitative data (i.e., input values defining building geometries and output values representing performance results) and observing qualitative data (i.e., images showing building geometries).

Based on the above idea, a Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method suitable for use in ill-defined conceptual architectural design has been proposed. This method is the main output of this research. It incorporates knowledge-supported, dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). The incorporation of such re-formulation is the main innovation of this method, which differentiates this method from other methods in the field of architectural design optimization. Given that this method is especially meaningful for complex buildings, this research focuses on indoor sports halls which are typical examples of complex buildings.

The proposed method consists of three phases: Phase-I: Optimization Problem Initial-Formulation (Initial-OPF) - a phase responsible for "*formulating*" an initial multi-objective optimization problem; Phase-II: Optimization Problem Re-Formulation (Re-OPF) - a phase responsible for "*re-formulating*" previous multi-objective optimization problems; Phase-III: Optimization Problem Solving (OPS) - a phase responsible for "*solving*" a final multi-objective optimization problem. Among these phases, the re-formulation phase is the key one; and it consists of three groups of actions that can iterate multiple times (i.e., data generation, information and knowledge extraction, multi-objective optimization model re-formulation). Depending on the different numbers of iterations, two subtypes of the proposed method are distinguished. Subtype-I, namely Non-dynamic, Interactive Re-formulation method, includes one re-formulation iteration; Subtype-II, namely Dynamic, Interactive Re-formulation method, includes multiple re-formulation iterations. The first subtype is more suitable for the design context where the main purpose is to reduce existing design possibilities (i.e., shrink exploration space), such as the circumstance during the relatively late sub-phase of conceptual architectural design. The second subtype is more suitable for the design context where the main purpose is to spark new design possibilities (i.e., expand exploration space), such as the circumstance during the relatively early sub-phase of conceptual architectural design.

To support the implementation of the proposed method, a software workflow has been developed. This software workflow is the secondary output of this research. It integrates McNeel's Grasshopper, ESTECO's modeFRONTIER, and simulation software tools Daysim, EnergyPlus, and Karamba3D (that are embedded in Grasshopper). The integration relies on a newly developed integration plug-in called Grasshopper-modeFRONTIER (Gh-mF) node. The development of this new plug-in is based on the collaboration between the Chair of Design Informatics at TU Delft and ESTECO SpA. The author of the thesis has participated in the plug-in development and testing process.

To demonstrate the use of the proposed method and verify the benefits and associated affecting factors, two case studies concerning the conceptual design of indoor sports halls have been conducted. Case Study I considers the conceptual design of the overall geometry of a sports competition hall, in a context that highlights reducing existing design possibilities. Specifically, the geometric design variables of the roofs, skylights, external shadings, roof structures, and grandstands have been manipulated based on architectural, daylighting, structural, energy, and thermal performance criteria. The Subtype-I method (i.e., non-dynamic method) has been applied in this case study, focusing on a one-time re-formulation process that concerns mainly removing existing variables (i.e., refining an existing concept convergently). Case Study II considers the conceptual design of the skylight geometry of a sports training hall, in a context that highlights sparking new design possibilities. Specifically, the geometric design variables of the roofs, skylights, and internal shadings have been manipulated based on architectural, daylighting, energy, and cost performance criteria. The Subtype-II method (i.e., dynamic method) has been applied in this case study, focusing on a three-time re-formulation process that concerns mainly adding new variables (i.e., enriching new concepts divergently). These case studies have confirmed the benefits of adopting the proposed method, relative to adopting traditional methods that do not incorporate the re-formulation phase; on the other hand, they have revealed some factors affecting the behaviors of the proposed method.

Finally, at the end of the thesis, the main contributions of this research have been summarized; comprehensive answers to all research questions have been presented; the main limitations of this research and future research directions have been provided.

KEYWORDS Performance-Based Building Design, Optimal-Design Paradigm, Design as Exploration; Optimization Problem Re-Formulation, Multi-Objective Optimization, Multi-Disciplinary Optimization, Indoor Sports Halls

Samenvatting

Ontwerpen is een doelgerichte activiteit. De doelstellingen ervan worden vaak uitgedrukt in de vorm van “prestaties”, die aangeven hoe goed of slecht een ontwerp werkt. Prestatiegericht ontwerpen (PBD) is dus een natuurlijke keuze om de kwaliteit van een ontwerp te verbeteren. Op het gebied van het ontwerpen van gebouwen hebben enkele bekende architectenbureaus gekozen voor een op prestaties gebaseerd ontwerp (PBBD) dat relevante voordelen oplevert, hoewel deze benaderingen niet door de hele architectengemeenschap worden gebruikt. Grote verbeteringen van deze benaderingen zijn optimaal ontwerp-paradigma’s (d.w.z. paradigma’s waarbij het ontwerp van een systeem wordt geformuleerd of gedeeltelijk geformuleerd als een optimalisatieprobleem). Onlangs hebben deze verbeterde benaderingen hun weg gevonden naar conceptueel architectuurontwerp en een groeiend aantal ontwerpers in staat gesteld ervan te profiteren. Met name de verbeterde benaderingen van multidisciplinaire optimalisatie (MDO) en multi-objectieve optimalisatie (MOO) krijgen steeds meer aandacht, gezien hun vermogen om tijdens het conceptuele ontwerp van complexe gebouwen om te gaan met meerdere prestatiecriteria uit verschillende ontwerpdisciplines. Toch ontbreekt het bij de huidige optimalisatie-ontwerp paradigma’s vaak aan een manier om een accuraat optimalisatieprobleem te realiseren, wat ondanks het gebruik van geavanceerde optimalisatie-algoritmen betrouwbare ontwerpoplossingen in de weg staat. Dit is het belangrijkste probleem dat in dit onderzoek wordt behandeld. Het bereiken van een dergelijke manier is vooral belangrijk voor conceptueel architectuurontwerp, omdat in deze fase de ontwerptaken meestal slecht gedefinieerd zijn en de geformuleerde optimalisatieproblemen meestal slecht gestructureerd zijn. Het kan ontwerpers helpen meer betrouwbare conceptuele oplossingen te bereiken en beter geïnformeerde vroege beslissingen te nemen, zodat zij meer kunnen profiteren van de verbeterde Performance-Based Building Design (PBBD) benaderingen.

Een veelbelovende manier om het bovenstaande probleem aan te pakken is het belang van herformulering van het optimalisatieprobleem (Re-OPF). Voor conceptueel architectuurontwerp is de formulering van een optimalisatieprobleem eigenlijk essentiëler dan de oplossing ervan. Met andere woorden, een ideaal paradigma voor optimaal ontwerp zou de prioriteit moeten verschuiven van het oplossen van optimalisatieproblemen (OPS) naar het formuleren van optimalisatieproblemen (OPF), meer bepaald naar dynamische en interactieve

herformulering van optimalisatieproblemen (Re-OPF). Met een dergelijke herformulering kunnen ontwerpers doelstellingen, beperkingen en ontwerpvariabelen toevoegen en/of verwijderen (d.w.z. de doelstellingen- en ontwerpruimte verschuiven om onontgonnen gebieden op te nemen en/of bestaande gebieden uit te sluiten). De herformulering kan dus worden gezien als een nuttige manier om tot een betrouwbaarder optimalisatieprobleem te komen. Verder is het voor een beter geïnformeerde herformulering van cruciaal belang dat relevante kennis wordt geëxtraheerd. De kennis omvat, maar is niet beperkt tot, welke doelstellingen en beperkingen als zinvoller moeten worden beschouwd, welke ontwerpvariabelen als kansrijker moeten worden beschouwd, en hoe deze elementen op elkaar inwerken. De kennis kan niet alleen nuttig zijn voor convergente herformulering, maar ook voor divergente herformulering. Zij kan worden afgeleid door het analyseren van kwantitatieve gegevens (d.w.z. inputwaarden die gebouwgeometrieën definiëren en outputwaarden die prestatieresultaten weergeven) en het observeren van kwalitatieve gegevens (d.w.z. beelden die gebouwgeometrieën tonen).

Op basis van het bovenstaande idee is een Multi-Objectieve en Multi-Disciplinaire Optimalisatie (MOMDO)-methode voorgesteld die geschikt is voor gebruik bij ongedefinieerde conceptuele architectuurontwerpen. Deze methode is het belangrijkste resultaat van dit onderzoek. Ze omvat kennisondersteunde, dynamische en interactieve herformulering van optimalisatieproblemen (Re-OPF). De integratie van een dergelijke herformulering is de belangrijkste innovatie van deze methode, die deze methode onderscheidt van andere methoden op het gebied van architectonische ontwerpoptimalisatie. Aangezien deze methode vooral zinvol is voor complexe gebouwen, richt dit onderzoek zich op indoor sporthallen, die typische voorbeelden zijn van complexe gebouwen.

De voorgestelde methode bestaat uit drie fasen: Fase-I: Optimization Problem Initial-Formulation (Initial-OPF) - een fase die verantwoordelijk is voor het “formuleren” van een initieel multi-objectief optimalisatieprobleem; Fase-II: Optimization Problem Re-Formulation (Re-OPF) - een fase die verantwoordelijk is voor het “herformuleren” van eerdere multi-objectieve optimalisatieproblemen; Fase-III: Optimization Problem Solving (OPS) - een fase die verantwoordelijk is voor het “oplossen” van een definitief multi-objectief optimalisatieprobleem. Van deze fasen is de herformuleringsfase de belangrijkste; deze bestaat uit drie groepen van acties die meerdere iteraties kunnen doorlopen (d.w.z. gegevensgeneratie, informatie- en kennisextractie, herformulering van het multi-objectieve optimalisatiemodel). Afhankelijk van de verschillende aantallen iteraties, worden twee subtypes van de voorgestelde methode onderscheiden. Subtype-I, namelijk de niet-dynamische, interactieve herformuleringsmethode, omvat één herformuleringsiteratie; Subtype-II, namelijk de dynamische, interactieve herformuleringsmethode, omvat meerdere

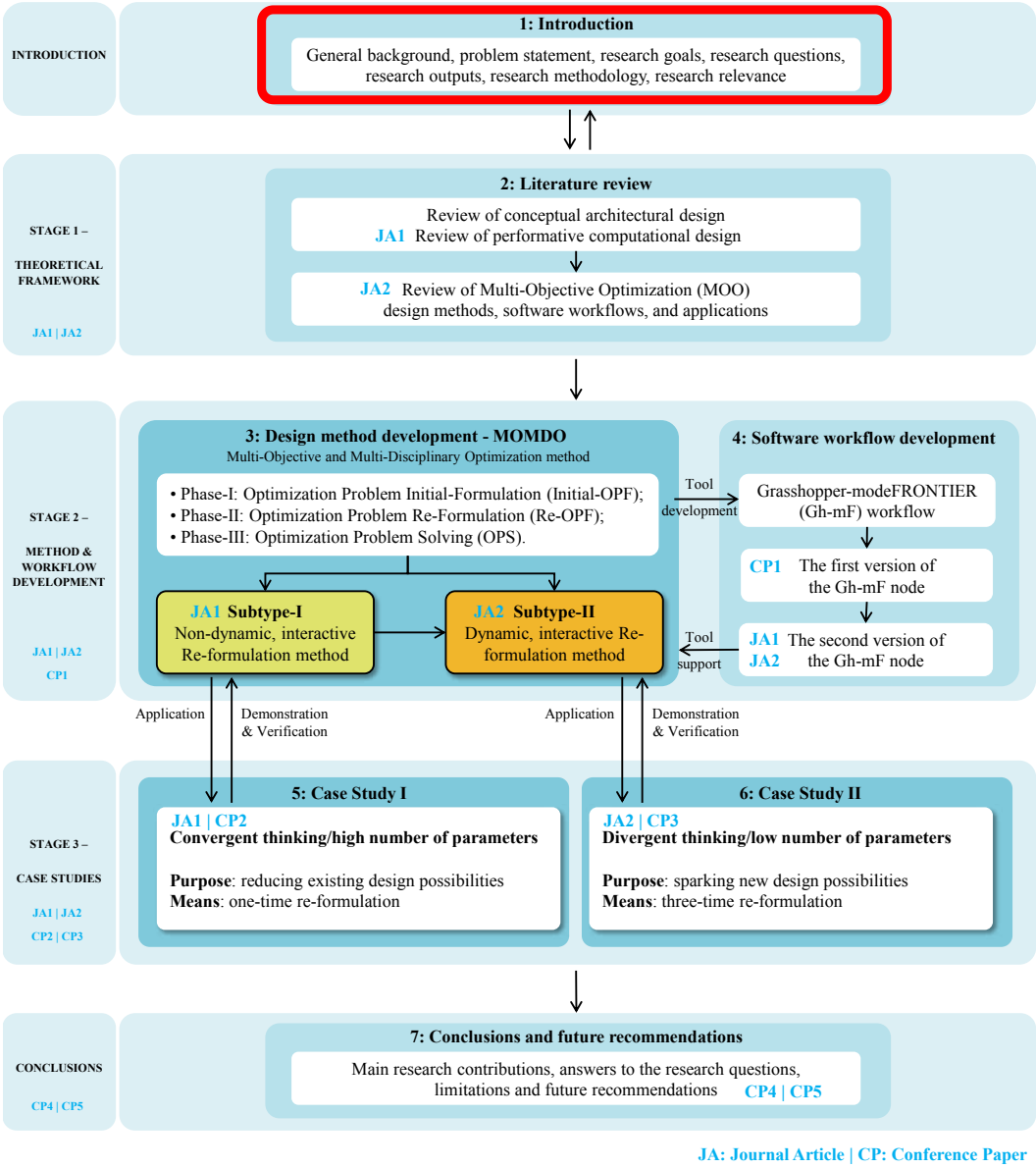
herformuleringsiteraties. Het eerste subtype is meer geschikt voor de ontwerpcontext waarin het hoofddoel is de bestaande ontwerpmogelijkheden te verminderen (d.w.z. de exploratieruimte te verkleinen), zoals de omstandigheid tijdens de relatief late subfase van conceptueel architectonisch ontwerp. Het tweede subtype is meer geschikt voor de ontwerpcontext waarin het hoofddoel is om nieuwe ontwerpmogelijkheden aan te boren (d.w.z. de exploratieruimte uit te breiden), zoals de omstandigheid is tijdens de relatief vroege subfase van conceptueel architectonisch ontwerp.

Ter ondersteuning van de implementatie van de voorgestelde methode is een software-workflow ontwikkeld. Deze software-workflow is het secundaire resultaat van dit onderzoek. Het integreert Grasshopper van McNeel, modeFRONTIER van ESTECO, en simulatiesoftwaretools Daysim, EnergyPlus, en Karamba3D (die zijn ingebed in Grasshopper). De integratie berust op een nieuw ontwikkelde integratie plug-in genaamd Grasshopper-modeFRONTIER (Gh-mF) node. De ontwikkeling van deze nieuwe plug-in is gebaseerd op de samenwerking tussen de leerstoel Design Informatics van de TU Delft en ESTECO SpA. De auteur van het proefschrift heeft deelgenomen aan de ontwikkeling en het testproces van de plug-in.

Om het gebruik van de voorgestelde methode te demonstreren en de voordelen en bijbehorende beïnvloedende factoren te verifiëren, zijn twee casestudies uitgevoerd betreffende het conceptuele ontwerp van indoor sporthallen. Casus I betreft het conceptuele ontwerp van de algemene geometrie van een sporthal, in een context waarin de bestaande ontwerpmogelijkheden worden beperkt. Meer bepaald zijn de geometrische ontwerpvariabelen van de daken, dakramen, externe zonwering, dakstructuren en tribunes gemanipuleerd op basis van architecturale, daglicht-, structurele, energie- en thermische prestatiecriteria. De Subtype-I methode (d.w.z. de niet-dynamische methode) is in deze casestudy toegepast, waarbij de nadruk ligt op een eenmalig herformuleringsproces dat voornamelijk betrekking heeft op het verwijderen van bestaande variabelen (d.w.z. het convergent verfijnen van een bestaand concept). Case Study II beschouwt het conceptuele ontwerp van de geometrie van het dakraam van een sporttrainingshal, in een context die nieuwe ontwerpmogelijkheden benadrukt. Meer bepaald werden de geometrische ontwerpvariabelen van de daken, dakramen en interne zonwering gemanipuleerd op basis van architecturale, daglicht-, energie- en kostencriteria. De Subtype-II methode (d.w.z. de dynamische methode) is in deze casestudy toegepast, waarbij de nadruk ligt op een drievoudig herformuleringsproces dat voornamelijk betrekking heeft op het toevoegen van nieuwe variabelen (d.w.z. het divergeren van nieuwe concepten). Deze casestudies hebben de voordelen van de voorgestelde methode bevestigd ten opzichte van de traditionele methoden die de herformuleringsfase niet omvatten; anderzijds hebben zij enkele factoren aan het licht gebracht die het gedrag van de voorgestelde methode beïnvloeden.

Ten slotte zijn aan het eind van het proefschrift de belangrijkste bijdragen van dit onderzoek samengevat; zijn uitgebreide antwoorden op alle onderzoeksvragen gepresenteerd; zijn de belangrijkste beperkingen van dit onderzoek en toekomstige onderzoeksrichtingen aangegeven.

SLEUTELWOORDEN Prestatiegericht Gebouw Ontwerp, Optimaal-Ontwerp Paradigma, Ontwerp als Verkenning; Optimalisatie Probleem Herformulering, Multi-Objectieve Optimalisatie, Multi-Disciplinaire Optimalisatie, Indoor Sporthallen



JA: Journal Article | CP: Conference Paper

TOC

1 Introduction

This chapter provides an overview of this research. First, it provides the general background (Section 1.1) and the problem statement (Section 1.2). Then, it describes the research goals, research questions, research outputs, and research methodology, respectively (Section 1.3, 1.4, 1.5 and 1.6); Finally, it describes the scientific and societal relevance of the research (Section 1.7).

1.1 General background

According to Tong and Sriram (1992), design is the process of constructing a description of an artifact that satisfies a set of given requirements. Many of these requirements can be expressed in the form of "*performance*". According to the Oxford dictionary, the term "*performance*" means how well or badly something works. Here it means how well or badly a building works in the field of building design. Performance-Based Building Design (PBBD) approaches are promising to improve the quality of a building design. In these approaches, various performance requirements are translated and integrated into a building design (Spekkink, 2005).

This research is concerned with Performance-Based Building Design (PBBD) approaches in general. To allow more designers to benefit from these approaches, this research advocate enhancing these approaches for conceptual architectural design by optimal-design paradigms. Thus, this section provides backgrounds concerning conceptual architectural design and optimal-design paradigms respectively (Section 1.1.1 and 1.1.2).

1.1.1 Background concerning conceptual architectural design

What is a conceptual architectural design and why is it important?

Design in architecture is a goal-directed activity in which decisions are taken about the physical form of buildings and their components to ensure their fitness for intended purposes, as defined by Radford and Gero (1980). According to their definition, a design task in architectural design contains at least two important components: the *goals* and the *means*.

Conceptual architectural design (or conceptual design for short) here refers to the early phase of architectural design where the goal is to meet meaningful design requirements by proposing promising design concepts. Thus, the two important components of a conceptual design task are the *design requirements* and the *design concepts*.

Conceptual architectural design is important, mainly because decisions made in this phase often have significant impacts on the success of a design. These decisions include those concerning the choice of design requirements and concepts. According to Pahl et al. (2007), a successful design solution is more likely to derive from the choice of the most appropriate concepts than from exaggerated concentration on technical details, as it is extremely difficult or impossible to correct fundamental shortcomings of early concepts in the late detailed design phase. Moreover, the choice of the most meaningful design requirements is even more important, as it can significantly affect the choice of design concepts. Thus, it is important to focus more on the conceptual design phase.

What are design knowledge and its classification?

The notion of *knowledge* is closely associated with the notions of *information* and *data*. In this research, data means values and images, including quantitative data (i.e., input values defining building geometries and output values representing performance results) and qualitative data (i.e., images showing building geometries); information means the processed data that is given meaning in a context; and knowledge means the organized information that can be put into practice in some way.

To be more specific, *design knowledge* here not only refers to the knowledge of various disciplines that can be applied in conceptual architectural design but also the knowledge of structuring and re-structuring existing disciplinary knowledge.

Moreover, there are also other meaningful classifications. According to Tong and Sriram (1992), design knowledge can be classified into *convergent knowledge* – the knowledge for helping a design process to converge on an acceptable design solution; and *divergent knowledge* – the knowledge for generating new design solutions in design space. According to the types of associated performances, design knowledge can be classified into *quantitative knowledge* – the knowledge about quantitative performance (e.g., climatic, structural, and energy performances whose evaluations primarily rely on complex mathematical calculations or simulations), and *qualitative knowledge* – the knowledge about qualitative performance (e.g., aesthetics performance whose evaluation primarily relies on human subjectivity).

The necessity of increasing knowledge in conceptual architectural design

It is necessary to increase design knowledge during conceptual architectural design. This necessity can be seen from the paradox between design knowledge and design freedom (see FIG.1.1). This paradox means that as a designer increases his/her knowledge about a design, he/she may lose the freedom to act on that knowledge (La Rocca, 2011); or in short, the more you learn the less freedom you have to use what you know (Ullman, 2010). It is especially true in the conceptual design phase. As illustrated by the solid curves in FIG.1.1, traditionally, the conceptual design phase has a higher level of design freedom that decreases rapidly, but a lower level of design knowledge that increases slowly, compared with the late design phases (Schrage et al., 1991). As illustrated by the dashed curves in FIG.1.1, ideally, the design knowledge can be increased, which can allow for more design freedom to support better design decision-making in the conceptual design phase (Mavris and DeLaurentis, 2000).

The necessity to increase design knowledge exists for both novice and expert architects and engineers. For novices, they may have shortcomings in terms of general disciplinary knowledge. In this circumstance, they have the opportunity to learn such knowledge by studying the data of alternative designs during conceptual architectural design. For experts, it is not always easy to determine which design requirements should be considered more meaningful, which design concepts should be considered more promising, and how the design requirements and concepts interact with each other, especially when dealing with complex projects in a multidisciplinary design environment. They may need to re-structure their existing disciplinary knowledge and be clearer about how to apply it to specific cases. In this sense, they can also acquire relevant knowledge from the study of the data.

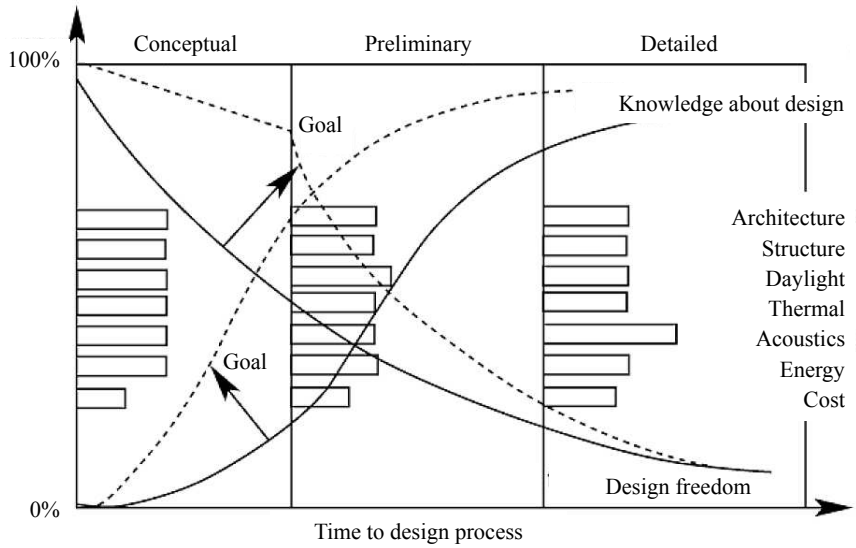


FIG. 1.1 The paradox between design knowledge and design freedom in a design process (revised from Schrage et al., 1991)

Note: The solid and dashed curves respectively represent the paradox in a traditional and ideal design process. The horizontal bars show the distribution of disciplines across the main phases of a design process.

The difficulty of extracting knowledge in conceptual architectural design

Knowledge extraction is often difficult in conceptual architectural design. The dynamic changes of design goals and other factors can make knowledge extraction difficult. As explained by Radford and Gero (1980): "*In order to take design decisions, the architect needs information (and knowledge) on the relationship between his goals and the means … (But it is) difficult enough to provide (such information and knowledge), because in architectural design (especially conceptual architectural design) little is static or permanent … It may not be possible to completely state his goals; other factors may come into play.*"

The difficulty of knowledge extraction can be obviously upgraded, when it comes to the conceptual design of complex buildings. The building complexity can bring significant challenges to knowledge extraction. This research focuses on indoor sports halls as an example. This type of building is complex in terms of both geometries and performance. On one hand, indoor sports halls can have a large number of building components that are assembled in regular or irregular ways;

TOC

and the resulting geometries may have complex impacts on building performances. On the other hand, they often include sophisticated performance measures from multiple disciplines; and these measures may conflict with each other to varying degrees. In this circumstance, design knowledge becomes significantly complex, which makes it difficult to be extracted.

The ill-defined nature of design tasks in conceptual architectural design

In this thesis, we use the term *ill-defined* or *inaccurate* to describe a design task that has no definitive and reliable formulation of the problem and solutions to the problem (Zhu, 2005). In other words, an ill-defined or inaccurate design task here refers to a design task that contains uncertain (soft and hard) design requirements and/or uncertain design concepts.

Design tasks are usually ill-defined, especially in conceptual architectural design where design knowledge is insufficient and knowledge extraction is difficult. Simon (1973) stated that a building design task is often ill-defined in many aspects: the design requirements are usually vague or unknown; the design concepts in terms of forms, structures, and materials can have various valid options; and the design process can be organized in various ways. Moreover, as agreed by many studies, most real-world design tasks are actually ill-defined, especially in the conceptual design phase (Smithers and Troxell, 1990; Navinchandra, 1991; Smithers, 1992; Logan and Smithers, 1993; Jonas, 1993; Gero, 1994; Smithers et al., 1994; Goel, 1995; Maher et al., 1996; Maher and Poon, 1996; Goldschmidt, 1997). The ill-defined nature of design tasks is not good for obtaining reliable design solutions.

1.1.2 Background concerning optimal-design paradigms

What is an optimal-design paradigm and why is it needed?

A paradigm refers to a typical example or pattern of something, according to the Oxford dictionary; and it can also refer to a set of assumptions, concepts, values, and practices that constitute a way of viewing reality, according to the American Heritage dictionary.

An optimal-design paradigm (or interchangeably a design optimization paradigm) here refers to a paradigm where the design of a system is formulated, or partially formulated, as a problem of optimization (Arora, 2016). Here, an optimization problem or model contains at least three important components: objectives, constraints, and design variables. Specifically, an optimization problem is a problem of minimizing or maximizing objectives and fulfilling constraints by manipulating the values of design variables, whereas an optimization model is a mathematical representation of an optimization problem.

Optimal-design paradigms are needed, mainly because they can enhance Performance-Based Building Design (PBBD) approaches. Optimal-design paradigms can be combined with parametric modeling and performance simulation. By doing so, Performance-Based Building Design (PBBD) approaches are enhanced with intelligent optimization algorithms, thus facilitating the search for optimal solutions. The approaches enhanced by the above computational means are also called *performative computational design* approaches, as proposed by Prof. Sevil Sariyildiz (2012) and exemplified in Turrin et al. (2011).

What are Multi-Disciplinary Optimization (MDO) and Multi-Objective Optimization (MOO)?

Multi-Disciplinary Optimization (MDO) (see FIG.1.2) and Multi-Objective Optimization (MOO) (see FIG.1.3) are two important optimal-design paradigms. They were originally used to deal with multi-disciplinary and multi-objective engineering design tasks and were later found useful for handling architectural design tasks (Evins, 2013; Nguyen et al., 2014; Huang and Niu, 2016). They have different focuses and advantages, as described below.

Multi-Disciplinary Optimization (MDO) focuses on how to collaboratively optimize the performances of different sub-systems or disciplines of a complex design system such as an aerospace vehicle. Its main advantage is that it is more likely to obtain better optimal solutions by handling the couplings between different sub-systems or disciplines than by optimizing each sub-system or discipline sequentially (Chittick and Martins, 2009). Since the 1990s, Multi-Disciplinary Optimization (MDO) has formally become an important field of research (Schrage et al., 1991; Sobieszczanski-Sobieski, 1995; Giesing and Barthelemy, 1998). It was initially applied in the field of aircraft design and was later applied to more engineering design fields, including the field of automobile design, train design, ship design, bridge design, etc., given the multi-disciplinary complexity of these design systems (Martins and Lambe, 2013).

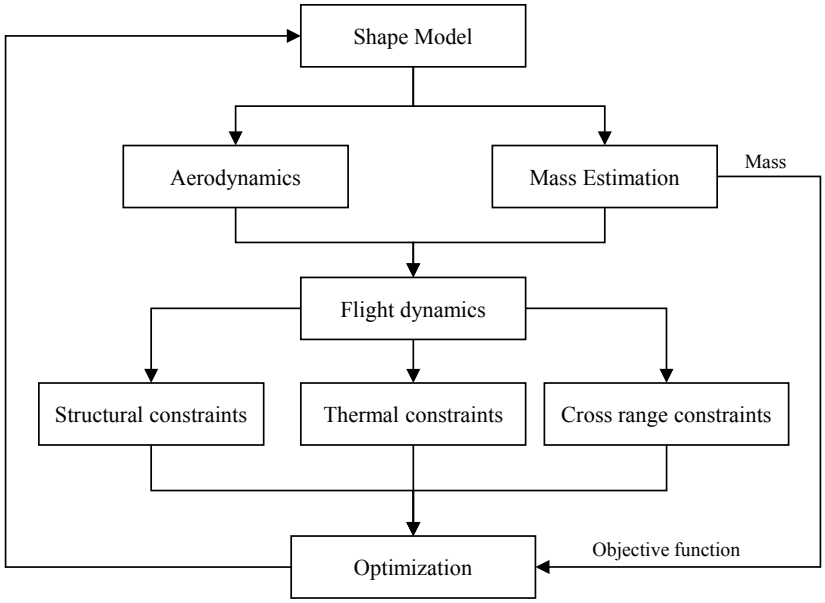


FIG. 1.2 Multi-Disciplinary Optimization (MDO) – an example of optimizing an aerospace vehicle (revised from Viviani et al, 2017)

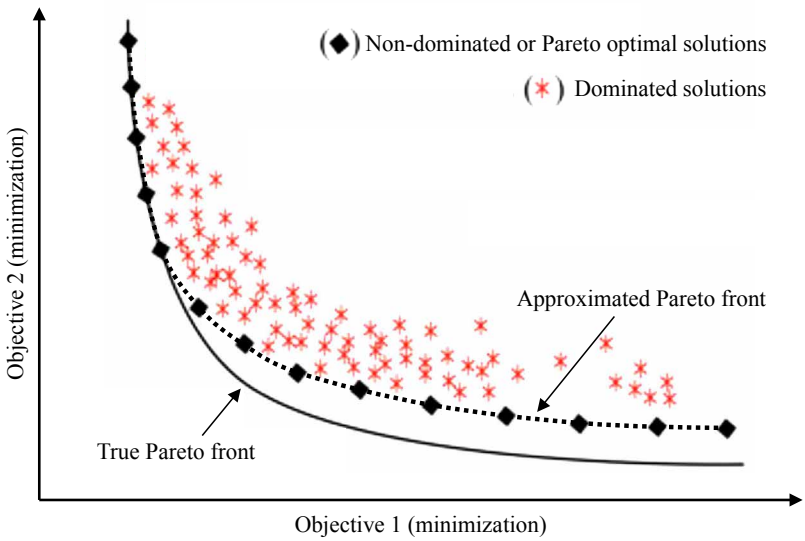


FIG. 1.3 Multi-Objective Optimization (MOO) (revised from Santín et al., 2017)

Note: For Pereto optimal solutions, no one objective function can be improved without a compromise in at least one of the other objectives.

TOC

Multi-Objective Optimization (MOO) focuses on how to simultaneously optimize multiple conflicting and incommensurable performance objectives. Its main advantage is that designers can understand the best trade-off between multiple objectives with the help of a set of non-dominated solutions (i.e., Pareto optimal solutions) obtained. Designers can select preferred solutions for further development from Pareto optimal solutions, based on their subjective preferences. This is actually a particular type of Multi-Objective Optimization (MOO), known as a posterior preference articulation approach (Andersson, 2000). In this approach, Pareto optimal solutions are independent of designers' preferences, thus optimization operations only need to be performed once (Andersson, 2000). A widely used Multi-Objective Optimization (MOO) algorithm is the Non-dominant Sorting Genetic Algorithm II (NSGA-II) proposed by Deb et al. (2002).

The combination of Multi-Disciplinary Optimization (MDO) and Multi-Objective Optimization (MOO)

The idea of combining Multi-Disciplinary Optimization (MDO) and Multi-Objective Optimization (MOO) is natural, given the above advantages. This idea has been well-developed in the field of engineering design. It has also got increasing attention recently in the field of architectural design, especially in the conceptual design phase, as shown in recent reviews (Østergård et al., 2016; Touloupaki and Theodosiou; 2017; Ekici et al., 2019).

This research is interested in the Multi-Objective and Multi-Disciplinary Optimization (MOMDO) paradigm which is the area where Multi-Disciplinary Optimization (MDO) and Multi-Objective Optimization (MOO) intersect. Nevertheless, this research also keeps an eye on the area that is beyond that intersection but within the scope of Multi-Objective Optimization (MOO), due to the lack of relevant studies.

The framework of the existing Multi-Objective and Multi-Disciplinary Optimization (MOMDO) paradigm

The existing Multi-Objective and Multi-Disciplinary Optimization (MOMDO) paradigm has a framework consisting of three basic modules: a geometry generation module, a performance analysis module, and an optimization module, as illustrated in FIG.1.4. In this framework, multi-disciplinary performance analysis and multi-objective optimization are involved.

This framework can be divided into two phases: Optimization Problem Formulation (OPF) and Optimization Problem Solving (OPS). In the formulation phase, a design task is converted into an optimization problem; more specifically, design requirements are converted into performance objectives and constraints, and design concepts are converted into design variables. In the solving phase, the geometry generation module is responsible for creating geometric parametric models; the (multi-disciplinary) performance analysis module can contain various analytical calculations or numerical simulations from different sub-systems or design disciplines, and the (multi-objective) optimization module can adopt different Multi-Objective Optimization (MOO) algorithms belonging to the posterior preference articulation approach.

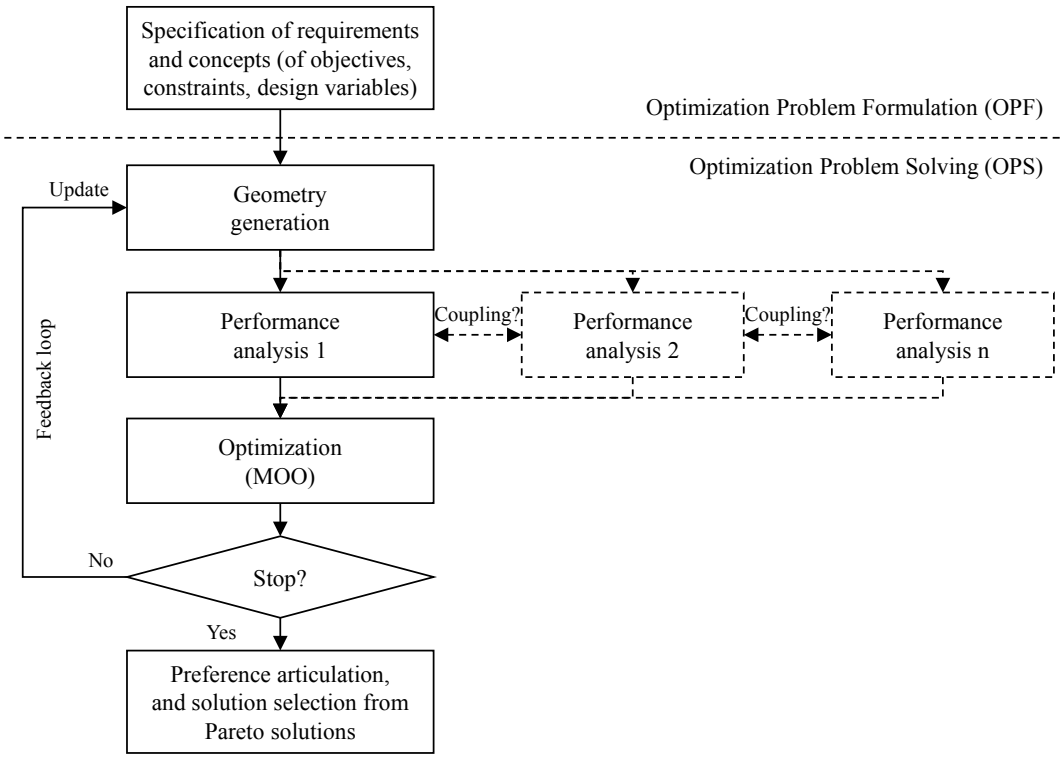


FIG. 1.4 A framework that integrates a geometry generation module, a (multi-disciplinary) performance analysis module, and a (multi-objective) optimization module

Overall, this framework can help designers make performance-based design decisions efficiently during conceptual architectural design, through feedback loops from performance assessment to design modification.

The ill-structured nature of optimization problems in an optimal-design paradigm

Similar to the terms that are used to describe a design task (in p. 39), we use the term *ill-structured* or *inaccurate* to describe an optimization problem that has no definitive and reliable objectives, constraints, and design variables. In other words, an ill-structured or inaccurate optimization problem here refers to an optimization problem that contains uncertain (qualitative and quantitative) objectives and constraints and/or uncertain design variables.

Optimization problems are usually ill-structured, given that they are just the approximation of design tasks and unavoidable inaccuracies exist. On one hand, the inaccuracies of the optimization problem can be inherited from the ill-defined nature of the design task. On the other hand, they can also come from other sources such as the approximation of hard-to-quantify criteria, the combination of multiple criteria into one, and the limited time for the possible refinements of the optimization problem, according to Meignan et al. (2015). In this sense, even though a design task is accurate or well-defined, the corresponding optimization problem can still be inaccurate or ill-structured. The ill-structured nature of optimization problems is not good for obtaining reliable design solutions.

1.2 Problem statement

As explained above, current optimal-design paradigms can enhance Performance-Based Building Design (PBBD) approaches to some degree. However, their limitations in supporting ill-structured optimization problems create a major barrier to obtaining reliable design solutions in conceptual architectural design. These limitations must be overcome to make more designers benefit from the enhanced Performance-Based Building Design (PBBD) approaches.

The main problem addressed in this research is: **in current Multi-Objective and Multi-Disciplinary Optimization (MOMDO) paradigms for ill-defined conceptual architectural design, there is often a lack of a way to ensure the achievement of a reliable optimization problem, which hinders reliable design solutions despite the use of advanced optimization algorithms.**

An important reason for this problem is that the potentials of **Optimization Problem Re-Formulation (OPF),** more precisely, **knowledge-supported, dynamic, and interactive re-formulation Optimization Problem Re-Formulation (Re-OPF),** have not been fully explored. For many architects and engineers who follow optimal-design paradigms, they tend to just focus on Optimization Problem Solving (OPS) without thoroughly discussing the reliability of the optimization problems in the first place. This obviously leads to a high risk of getting unreliable design solutions. For those who implement limited Optimization Problem Re-Formulation (Re-OPF), they tend to shrink objective space and design space to exclude existing areas, rather than expand these spaces to include new and unexplored areas. This may prevent them from getting more reliable and more divergent design solutions.

This problem is relevant for various building types, particularly for buildings like indoor sports halls (which are selected as an application field in this research). This type of buildings differs from other building types primarily by its multifunctional purposes and large-span space. It is often complex in terms of both performance requirements and design concepts. The performance requirements can come from multiple disciplines, including but not limited to sightline, seat number, clear height, visual and thermal comfort, structural safety and serviceability, embodied and operational energy. The design concepts proposed for handling these requirements can involve complex geometry. This complexity raises significant challenges for the conceptual design of indoor sports halls. In this design phase, relevant performance requirements and promising design concepts are not clear (i.e., the design task is ill-defined). Thus, the optimization problem initially formulated (from the ill-defined design task) is probably not reliable, which needs to be re-formulated and improved.

The issue regarding software workflows should also be addressed to help solve the main problem. According to RoboDK (2019), a software workflow refers to "*the definition, execution, and automation of software processes where tasks, information or documents are passed from one program to another for action, according to a set of procedural rules*". Most of the existing software workflows have not been designed for supporting Optimization Problem Re-Formulation (Re-OPF). Thus, they often have limitations in offering functions that are important for the re-formulation, such as user-friendly parametric geometric modeling, useful sampling algorithms, and handy post-processing tools.

1.3 Research goals

This research aims to achieve the main goal (Section 1.3.1) and several sub-goals that are directly or indirectly associated with the main goal (Section 1.3.2).

1.3.1 Main goal

The main goal of this research is: **to develop a Multi-Objective and Multi-Disciplinary Optimization method suitable for use in ill-defined conceptual architectural design, by leveraging information and knowledge extraction to support dynamic and interactive Optimization Problem Re-Formulation (Re-OPF).**

It is not the goal of this research to develop a rigid method that fully automates the design process and fully delegates human creativity to computational procedures. Instead, this research intends to develop a flexible method that partially automates the design process and considerably enhances human creativity through computational techniques. Such flexibility is desired to enable the method to adapt to different design contexts, and it can be derived from dynamic and interactive re-formulation.

1.3.2 Sub-goals

The main goal of this research is decomposed into four groups of sub-goals. The grouping of the sub-goals is based on their relations with the main goal.

The sub-goals of the first group are concerned with the theoretical framework of this research, and help achieve the main goal, namely:

1. To ascertain a way that can help to achieving a reliable design task and a reliable optimization problem and identify the general state of optimal-design methods in supporting this way.

2. To identify the state of the art of Multi-Objective Optimization (MOO) design methods, software workflows, and application to the conceptual design of sports buildings.

The sub-goal of the second group is concerned with an optimal-design method, and is directly associated with the main goal, namely:

3 To develop an optimal-design method that enables information and knowledge extraction and hence dynamic and interactive Optimization Problem Re-Formulation (Re-OPF).

The sub-goal of the third group is concerned with a software workflow, and helps achieve the main goal, namely:

4 To establish a software workflow that can support the implementation of the above optimal-design method.

The sub-goal of the fourth group is concerned with case studies, and helps achieve the main goal, namely:

5 To provide case studies that can be used to establish the validity of the above optimal-design method.

1.4 Research questions

This research raises the main question (Section 1.4.1) and several sub-questions that are directly or indirectly associated with the main question (Section 1.4.2).

1.4.1 Main question

The main question of this research is: **how to assist architects and engineers to extract useful information and knowledge to support dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) during ill-defined conceptual design.**

1.4.2 Sub-questions

The main question of this research is decomposed into four groups of sub-questions. The grouping of the sub-questions is based on their relationship with the main question.

The sub-questions of the first group are concerned with the theoretical framework of this research (Chapter 2), and help address the main question, namely:

1. How to improve the reliability of an design task and an optimization problem, and to what extent do current optimal-design methods deal with this issue?

2. To what extent: (1) is dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) supported? (2) are necessary computational techniques provided? (3) are optimal-design methods applied to the conceptual design of sports buildings?

The sub-question of the second group is concerned with the proposed optimal-design method (Chapter 3), and is directly associated with the main question, namely:

3. How to arrange actions and adopt necessary computational techniques for the proposed optimal-design method?

The sub-question of the third group is concerned with the proposed software workflow (Chapter 4), and helps address the main question, namely:

4. How to select software tools and integrate them seamlessly into the proposed software workflow?

The sub-question of the fourth group is concerned with the case studies (Chapters 5 and 6), and helps address the main question, namely:

5. How to demonstrate the use of the proposed optimal-design method and verify its benefits and associated affecting factors through case studies concerning indoor sports halls?

1.5 Research outputs

1.5.1 Main output

The main output of this research is: **a Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method suitable for use in ill-defined conceptual architectural design**. This method incorporates knowledge-supported, dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). The incorporation of such re-formulation is the main innovation of this method, which differentiates this method from other methods in the field of architectural design optimization.

This method consists of three phases:

- Phase-I: Optimization Problem Initial-Formulation (Initial-OPF);
- Phase-II: Optimization Problem Re-Formulation (Re-OPF);
- Phase-III: Optimization Problem Solving (OPS).

Moreover, this method contains two subtypes that differ from each other in the way of the re-formulation:

- Subtype-I: Non-dynamic, Interactive Re-formulation method;
- Subtype-II: Dynamic, Interactive Re-formulation method.

The Subtype-I method (i.e., non-dynamic method) includes one re-formulation iteration. It is more suitable for the design context where the main purpose is to reduce existing design possibilities (i.e., shrink exploration space), such as the circumstance in the relatively late sub-phase of conceptual architectural design.

The Subtype-II method (i.e., dynamic method) includes multiple re-formulation iterations. It is more suitable for the design context where the main purpose is to spark new design possibilities (i.e., expand exploration space), such as the circumstance in the relatively early sub-phase of conceptual architectural design.

1.5.2 Secondary output

The secondary output of this research is: **a software workflow developed to support the implementation of the proposed method**. This workflow integrates McNeel's Grasshopper, ESTECO's modeFRONTIER, and simulation software tools Daysim, EnergyPlus, and Karamba3D (that are embedded in Grasshopper).

1.6 Research methodology

In general, research methodology is a way to systematically solve the research problem. It is not only concerned with specific methods or techniques that are used to perform research operations, but also with the steps or processes for carrying out the research (Kothari, 2004).

This research adopted a three-stage process. The three research stages (i.e., Stage 1 – Stage 3) correspond to the three main parts of the thesis (i.e., theoretical framework, method and workflow development, and case studies) respectively, as illustrated in FIG.1.5. In these research stages, different research tasks were carried out by using different methods or techniques, as described below.

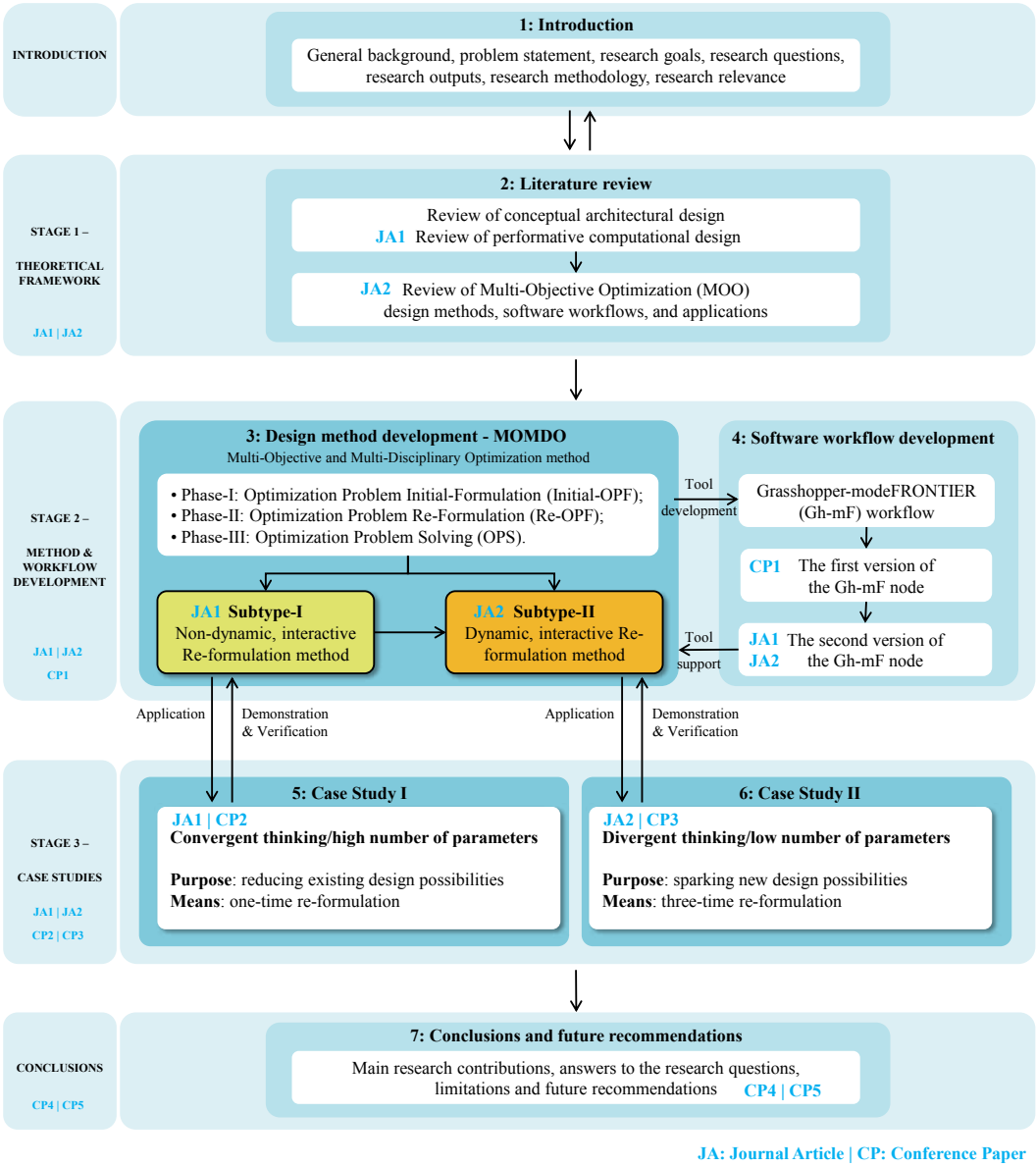


FIG. 1.5 The overview of the research methodology

Stage 1 - theoretical framework (Chapter 2)

Stage 1 aimed to identify research problems, and define research goals and questions. More specifically, this stage helps to have an overall picture of conceptual architectural design and performative computational design and further understanding of state-of-the-art Multi-Objective Optimization (MOO) design methods, software workflows, and applications. This information is useful for guiding the consequent development of the research. For this, systematic literature reviews and relevant interviews were conducted.

First, the theories of conceptual architectural design and performative computational design were reviewed, to know a potential means to improve the reliability of an design task and an optimization problem, and to understand the extent these potential means were discussed in existing optimal-design methods. Then, Multi-Objective Optimization (MOO) design methods, software workflows, and applications were further reviewed, to understand the extent (1) dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) were implemented, (2) necessary computational techniques were provided, and (3) optimal-design methods were applied to the conceptual design of sports buildings. Moreover, interviews concerning the above topics were also held with experts from the fields of architectural design, engineering design, and computational design.

Stage 2 - method and workflow development (Chapters 3 and 4)

Stage 2 aimed to create a desired optimal-design method and software workflow. For this, interdisciplinary research was conducted, which involves theories, concepts, techniques, and tools not only from architecture and building engineering disciplines, but also from statistics, computer science, and software engineering disciplines.

Based on the previous theoretical framework, a promising direction for developing the method was set. Following this direction, the method was proposed. It incorporated knowledge-supported, dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). According to different ways of the re-formulation, it was subdivided into two subtypes. The second subtype was derived by further extending the first subtype. The main purpose for distinguishing these two subtypes was to improve the flexibility of the method to accommodate different design contexts. These two subtypes were continuously adjusted based on the feedback of some simplified tests during their development processes. Moreover, computational techniques were a necessary complement to the method; they were adopted for supporting relevant actions of the method.

To provide computational techniques necessary for the method, a software workflow was needed. Based on the previous theoretical framework, a promising direction for developing the workflow was set. Following this direction, the workflow was developed. It integrated McNeel's Grasshopper, ESTECO's modeFRONTIER, and simulation software tools Daysim, EnergyPlus, and Karamba3D (that were embedded in Grasshopper). The integration relied on a newly developed integration plug-in called Grasshopper-modeFRONTIER (Gh-mF) node. The development of this node was based on the collaboration between the Chair of Design Informatics at TU Delft and ESTECO SpA. The author of the thesis participated in the development process together with ESTECO's interdisciplinary team. For creating the node, the author specified the desired software workflow as a whole, implemented relevant functions in Grasshopper, and engaged in brainstorming ideas for the software integration; for verifying the node, the author ran some internal tests and helped coordinate external testing activities like workshops. The development went through several rounds, thus producing at least two work-in-progress versions of the node; the second version led to the final node available in modeFRONTIER.

Stage 3 - case studies (Chapters 5 and 6)

Stage 3 aimed to demonstrate the use of the proposed method and verify the benefits and associated affecting factors. For this, two case studies concerning indoor sports halls were conducted, with the aid of the proposed software workflow and integration plug-in.

The two case studies were selected primarily because they were complementary in terms of design phases. Case Study I focused on the relatively late sub-phase of the conceptual design where convergent thinking is often highlighted or the number of parameters is usually high. In contrast, Case Study II focused on the relatively early sub-phase of the conceptual design where divergent thinking is often highlighted or the number of parameters is relatively low.

Given the above design contexts, the two subtypes of the proposed method were applied to the two case studies respectively. Case Study I demonstrated the use of the Subtype-I method (i.e., non-dynamic method), with a focus on a one-time re-formulation process that concerns mainly removing existing variables (i.e., refining an existing concept convergently). Case Study II demonstrated the use of the Subtype-II method (i.e., dynamic method), with a focus on a three-time re-formulation process that concerns mainly adding new variables (i.e., enriching new concepts divergently).

At the end of each case study, the benefits of adopting the related subtype method and the factors affecting its behaviors were verified internally by the author of the thesis using comparative analysis. In addition, reflections on each case study were conducted, which helped further the understanding of the related subtype method and provide valuable feedback about possible extensions or applications of the method.

A series of Journal Articles (i.e., JA1-JA2) and Conference Papers (i.e., CP1-CP5) in relation to this research have been published, as listed below:

- JA1: Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. Automation in Construction, 92, 242-269. DOI: https://doi.org/10.1016/j.autcon.2018.03.023.

- JA2: 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, 103251. DOI: https://doi.org/10.1016/j.autcon.2020.103251.

- CP1: Yang, D., Sun, Y., Turrin, M., Buelow, P. V., & Paul, J. (2015). Multi-objective and multidisciplinary design optimization of large sports building envelopes: a case study. In: Proceedings of IASS Annual Symposia, IASS 2015 Amsterdam Symposium: Future Visions – Computational Design (pp. 1-14). IASS. ISSN: 2518-6582.

- CP2: Yang, D., Turrin, M., Sariyildiz, S., & Sun, Y. (2015). Sports building envelope optimization using multi-objective multidisciplinary design optimization (M-MDO) techniques: Case of indoor sports building project in China. In: 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 2269-2278). IEEE, Piscataway. ISBN: 978-1-4799-7492-4.

- CP3: Yang, D., Sun, Y., di Stefano, D., & Turrin, M. (2017). A computational design exploration platform supporting the formulation of design concepts. In: M. Turrin, B. Peters, W. O'Brien, R. Stouffs, & T. Dogan (Eds.), 2017 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 35-42). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-88878-6.

- CP4: Yang, D., Sun, Y., Sileryte, R., D'Aquilio, A., & Turrin, M. (2016a). Application of surrogate models for building envelope design exploration and optimization. In: R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), 2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 11-14). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

- CP5: Yang, D., Sun, Y., Di Stefano, D., Turrin, M., & Sariyildiz, S. (2016b). Impacts of problem scale and sampling strategy on surrogate model accuracy: An application of surrogate-based optimization in building design. In: 2016 IEEE congress on evolutionary computation (CEC) (pp. 4199-4207). IEEE, Piscataway. ISBN: 978-1-5090-0623-6.

These articles and papers are related to different parts of the thesis, as illustrated in FIG.1.5. Each of the journal articles relates to all three main parts of the thesis. The conference paper CP1 mainly relates to software workflow development. The conference papers CP2-CP3 mainly relate to the two case studies. The conference papers CP4-CP5 relate to the use of surrogate-based optimization, which has been considered a valuable direction for future research at the end of the thesis.

1.7 Research relevance

1.7.1 Scientific relevance

The scientific contribution of this research lies in expanding knowledge about performative computational design in general. Specifically, it relates to a new Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method proposed for use in ill-defined conceptual architectural design.

This research generates knowledge about a new perspective for dealing with ill-structured optimization problems (i.e., incorporating knowledge-supported, dynamic and interactive Optimization Problem Re-Formulation in optimal-design methods). First, the method based on this knowledge can offer a way to help the achievement of a more reliable optimization problem and hence more reliable design solutions. Second, this method can be used to deal with different types of design tasks. The Subtype-I method (i.e., non-dynamic method) can suit the need of refining an existing concept convergently, while the Subtype-II method (i.e., dynamic method) can suit the need of exploring new concepts divergently. The both subtype methods are useful for improving the proximity, diversity and geometric variation appropriateness of Pareto fronts.

1.7.2 Societal relevance

Society, including building practitioners and the general public, can benefit from this research in different ways.

Building practitioners, especially architects and engineers, can make the best use of Performance-Based Building Design (PBBD) approaches with the aid of the proposed method and software workflow. The proposed method allows them to extract useful information and knowledge to support the re-formulation of optimization problems, and thus allows them to make more informed early decisions during conceptual architectural design. This is good for obtaining more reliable design solutions. Moreover, the proposed method and software workflow can also facilitate architects and engineers to work collaboratively in a multi-disciplinary design environment in an early design phase.

The general public, especially sports enthusiasts, can benefit from the sports halls designed by using the proposed method. The proposed method can help improve the overall performances of sports halls, namely finding the best compromise or balance between possibly conflicting performances such as architectural, daylight, thermal, energy, and structural performances. The improvement of these performances not only benefits the general public by providing pleasing buildings and saving resources but also benefits sports enthusiasts by offering high-performing sports facilities.

References

Andersson, J. (2000). A survey of multiobjective optimization in engineering design. Technical Report, Linktjping University, Sweden. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=7DDF-6CE2B9F8F2C42D13381870BF151E?doi=10.1.1.8.5638&rep=rep1&type=pdf [Accessed: 24-September-2021].

Arora, J. S. (2016). Introduction to Optimum Design. Fourth Edition. Academic Press, Cambridge. ISBN: 9780128008065.

Boshernitsan, M., & Downes, M. S. (2004). Visual programming languages: a survey. Report of Computer Science Division, University of California, U.S. http://citeseerx.ist.psu.edu/viewdoc/download?-doi=10.1.1.73.2491&rep=rep1&type=pdf [Accessed: 24-September-2021].

Chittick, I. R., & Martins, J. R. (2009). An asymmetric suboptimization approach to aerostructural optimization. Optimization and Engineering, 10(1), 133-152. DOI: https://doi.org/10.1007/s11081-008-9046-2.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. DOI: http://doi.org/10.1109/4235.996017.

Ekici, B., Cubukcuoglu, C., Turrin, M., & Sariyildiz, I. S. (2019). Performative computational architecture using swarm and evolutionary optimisation: A review. Building and Environment, 147, 356-371. DOI: https://doi.org/10.1016/j.buildenv.2018.10.023.

Evins, R. (2013). A review of computational optimisation methods applied to sustainable building design. Renewable and Sustainable Energy Reviews, 22, 230-245. DOI: https://doi.org/10.1016/j.rser.2013.02.004.

Flager, F., Welle, B., Bansal, P., Soremekun, G., & Haymaker, J. (2009a). Multidisciplinary process integration and design optimization of a classroom building. Journal of Information Technology in Construction (ITcon), 14(38), 595-612. http://www.itcon.org/2009/38 [Accessed: 24-September-2021].

Gero, J. S. (1994). Towards a model of exploration in computer-aided design. In: J. S. Gero, & E. Tyugu (Eds.), Formal Design Methods for CAD: Proceedings of the IFIP TC5/WG5.2 Workshop on Formal Design Methods for CAD (pp. 315–336). Elsevier Science, Amsterdam. ISBN: 978-0-444-81970-3.

Giesing, J., & Barthelemy, J. F. (1998). A summary of industry MDO applications and needs. In: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization (pp. 1-15). AIAA, Reston. DOI: https://doi.org/10.2514/6.1998-4737.

Goel, V. (1995). Sketches of thought. The MIT Press, Cambridge. ISBN 0-262-07163-0.

Goldschmidt, G. (1997). Capturing indeterminism: representation in the design problem space. Design Studies, 18(4), 441-455. DOI: https://doi.org/10.1016/S0142-694X(97)00011-2

Huang, Y., & Niu, J. L. (2016). Optimal building envelope design based on simulated performance: History, current status and new potentials. Energy and Buildings, 117, 387-398. DOI: https://doi.org/10.1016/j.enbuild.2015.09.025.

Jonas, W. (1993). Design as problem-solving? or: Here is the solution—What was the problem? Design Studies, 14(2), 157-170. DOI: https://doi.org/10.1016/0142-694X(93)80045-E

Kothari, C. R. (2004). Research methodology: Methods and techniques. New Age International Publishers, New Delhi. ISBN: 978-81-224-2488-1.

La Rocca, G. (2011). Knowledge based Engineering techniques to support aircraft design and optimization (doctoral dissertation). Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands.

Logan, B., & Smithers, T. (1993). Creativity and design as exploration. In: J. S. Gero, & M. L. Maher (Eds.), Modeling Creativity and Knowledge-based Creative Design (pp. 139-173). Lawrence Erlbaum Associates, Hillsdale. ISBN: 0-8058-1153-2.

Maher, M. L., & Poon, J. (1996). Modeling design exploration as co-evolution. Computer-Aided Civil and Infrastructure Engineering, 11(3), 195-209. DOI: https://doi.org/10.1111/j.1467-8667.1996.tb00323.x

Maher, M. L., Poon, J., & Boulanger, S. (1996). Formalising design exploration as co-evolution. In: J. S. Gero, & F. Sudweeks (Eds.), Advances in formal design methods for CAD: Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for CAD (pp. 3-30). Springer Science + Business Media, Dordrecht. ISBN: 978-1-4757-4428-6.

Martins, J. R., & Lambe, A. B. (2013). Multidisciplinary design optimization: a survey of architectures. AIAA Journal, 51(9), 2049-2075. DOI: https://doi.org/10.2514/1.J051895.

Mavris, D. N., & DeLaurentis, D. A. (2000). Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design. In: I. Grant (Ed.), CD-ROM proceedings of 22nd International Congress of Aeronautical Sciences (1.4.4). Optimage, Edinburgh. ISBN 0-9533991-2-5.

Meignan, D., Knust, S., Frayret, J. M., Pesant, G., & Gaud, N. (2015). A review and taxonomy of interactive optimization methods in operations research. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(3), 1-43. DOI: https://doi.org/10.1145/2808234.

Navinchandra, D. (2011). Exploration and Innovation in Design: Towards a Computational Model. Springer-Verlag, New York. ISBN: 978-1-4612-7802-3.

Nguyen, A. T., Reiter, S., & Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. Applied Energy, 113, 1043-1058. DOI: https://doi.org/10.1016/j.apenergy.2013.08.061.

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. (2007). Engineering Design: A Systematic Approach. Third Edition. Springer-Verlag, London. ISBN: 978-1-84628-318-5.

Radford, A. D., & Gero, J. S. (1980). On optimization in computer aided architectural design. Building and Environment, 15(2), 73-80. DOI: https://doi.org/10.1016/0360-1323(80)90011-6.

RoboDK (2019). How to Set Up a Strong, Streamlined Software Workflow. https://robodk.com/blog/streamlined-software-workflow/ [Accessed: 24-September-2021].

Santín, I., Pedret, C., & Vilanova, R. (2016). Control and Decision Strategies in Wastewater Treatment Plants for Operation Improvement. Springer, Cham. ISBN: 978-3-319-46366-7.

Sariyildiz, S. (2012). Keynote Speech: Performative Computational Design. In: ICONARCH-I "Architecture & Technology" International Congress: Proceedings Book (pp. 313-344). Selçuklu Municipality of Konya, Konya. ISBN: 978-975-448-203-4.

Schrage, D., Beltracchi, T., Berke, L., Dodd, A., Niedling, L., & Sobieszczanski-Sobieski, J. (1991). Current State of the Art on Multidisciplinary Design Optimization (MDO): an AIAA White Paper. AIAA, Washington D.C. ISBN 1-56347-021-7.

Simon, H. A. (1973). The structure of ill structured problems. Artificial Intelligence, 4(3-4), 181-201. DOI: https://doi.org/10.1016/0004-3702(73)90011-8.

Smithers, T. (1992). Design as exploration: puzzle-making and puzzle-solving. https://www.academia.edu/22991882/Design_as_Exploration_Puzzle-Making_and_Puzzle-Solving [Accessed: 24-September-2021].

Smithers, T., Corne, D., & Ross, P. (1994). On computing exploration and solving design problems. In: J. S. Gero, & E. Tyugu (Eds.), Formal Design Methods for CAD: Proceedings of the IFIP TC5/WG5.2 Workshop on Formal Design Methods for CAD (pp. 293-313). Elsevier Science, Amsterdam. ISBN: 978-0-444-81970-3.

Smithers, T., & Troxell, W. (1990). Design is intelligent behaviour, but what's the formalism? Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 4(2), 89-98. DOI: https://doi.org/10.1017/S0890060400002286.

Sobieszczanski-Sobieski, J. (1995). Multidisciplinary design optimization: an emerging new engineering discipline. In: J. Herskovits (Ed.), Advances in Structural Optimization (pp. 483-496). Springer, Dordrecht. ISBN: 978-0-7923-2510-9.

Spekkink, D. (2005). Performance Based Design of Buildings. Final Domain Report. CIB, Rotterdam.

Touloupaki, E., & Theodosiou, T. (2017). Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization - A review. Energies, 10(5), 637. DOI: https://doi.org/10.3390/en10050637.

Tong, C., & Sriram, D. (Eds.). (1992). Artificial Intelligence in Engineering Design: Volume III: Knowledge Acquisition, Commercial Systems, And Integrated Environments. Academic Press, London. ISBN: 0-12-660563-7.

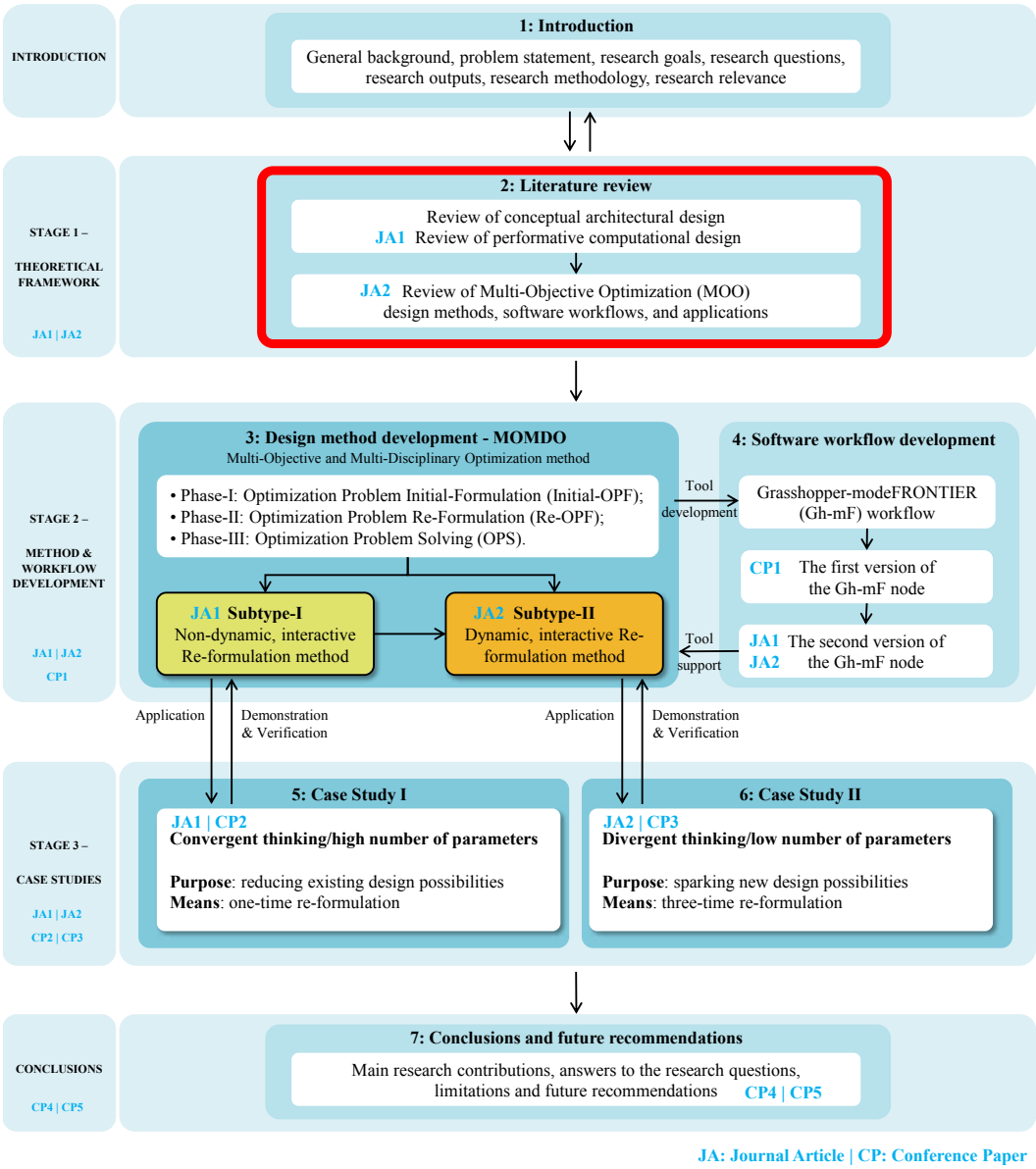
Turrin, M., Von Buelow, P., & Stouffs, R. (2011). Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. Advanced Engineering Informatics, 25(4), 656-675. DOI: https://doi.org/10.1016/j.aei.2011.07.009.

Ullman, D. G. (2010). The Mechanical Design Process. Fourth Edition. McGraw-Hill, New York. ISBN: 978-0-07-297574-1.

Viviani, A., Iuspa, L., & Aprovitola, A. (2017). An optimization-based procedure for self-generation of Re-entry Vehicles shape. Aerospace Science and Technology, 68, 123-134. DOI: https://doi.org/10.1016/j.ast.2017.05.009.

Zhu, H. (2005). Software Design Methodology: From Principles to Architectural Styles. Butterworth-Heinemann, Oxford. ISBN: 978-0-7506-6075-4.

Østergård, T., Jensen, R. L., & Maagaard, S. E. (2016). Building simulations supporting decision making in early design - A review. Renewable and Sustainable Energy Reviews, 61, 187-201. DOI: https://doi.org/10.1016/j.rser.2016.03.045.



JA: Journal Article | CP: Conference Paper

TOC

2 Literature review

This chapter reviews relevant areas of architectural design optimization and forms the theoretical framework that can guide the development of an optimal-design method and a software workflow.

The chapter is structured as follows. First, it introduces the purposes of the literature review (Section 2.1). Then, it provides basic concepts and reviews relevant topics in two fields: conceptual architectural design (Section 2.2) and performative computational design (Section 2.3). Next, it further reviews studies on Multi-Objective Optimization (MOO) design methods, software workflows, and applications respectively (Section 2.4, 2.5 and 2.6). Finally, it concludes by summarizing the main research results and providing concluding remarks (Section 2.7).

Sections 2.3-2.5 involve contents published in Journal Articles 1-2 (Yang et al., 2018; Yang et al., 2020).

2.1 Introduction

The purposes of the literature review are multifold. First of all, the review of the two fields is to identify a potential means to achieve a reliable design task in ill-defined conceptual architectural design, a potential means to achieve a reliable optimization problem in optimal design, and relevant challenges. Then, the review of Multi-Objective Optimization (MOO) design methods is to understand the state of the art in supporting dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). Next, the review of Multi-Objective Optimization (MOO) software workflows is to understand the state of the art in software selection and integration. Last, the Multi-Objective Optimization (MOO) application review is to understand the current trends in applying relevant design methods and software workflows to the conceptual design of indoor sports halls.

2.2 Conceptual architectural design

This section provides some different interpretations of conceptual design in different contexts (Section 2.2.1); then, it describes the importance of achieving a reliable design task in conceptual architectural design (Section 2.2.2); and last, it identifies a potential means to achieve a reliable design task (Section 2.2.3).

2.2.1 Interpretations of conceptual design

There is no consensus definition of conceptual design. Conceptual design can be interpreted differently, as it has different forms and goals in different domains or from other perspectives (Horváth, 2004). Given this fact, it is worth being aware of some different interpretations.

Conceptual design in industrial design

According to Pahl et al. (2007), a typical industrial design process can consist of four phases: task clarification, conceptual design, embodiment design, and detail design. In their definition, the conceptual design phase refers to the process where the basic solution path is laid down via the elaboration of a solution principle also called a concept. Moreover, according to Ulrich and Eppinger (2012), a typical industrial design process can also consist of six phases: planning, concept development, system-level design, detail design, testing and refinement, and production ramp-up. In their definition, the concept development phase refers to the process where the needs of the target market are identified, alternative product concepts are generated and evaluated, and one or more concepts are selected for further development and testing.

Conceptual design in architectural design

According to the American Institute of Architects (2017), an architectural design process generally consists of five phases: schematic design, design development, construction documents, procurement, and construction administration. Besides, according to Fontan (2021), Harpster (2021), and Schneider (2018), a phase known as pre-design or programming can be also added before the schematic design phase.

To the definition of the American Institute of Architects (2017), the schematic design phase is the process aiming to develop a preliminary design that illustrates the scale and relationship of the building components; typical deliverables of this phase can include preliminary site plans, building plans, sections and elevations, study models, perspective sketches, digital representations, preliminary selections of major building systems and materials. Given that the schematic design phase is fairly conceptual by nature (Schneider, 2018), it is sometimes called a conceptual design phase where various concepts are explored and narrowed down to one preferred concept.

Conceptual design from other perspectives

In addition to the above interpretations based on the subdivision of design processes, conceptual design can be also interpreted from other perspectives. According to Horváth (2004), conceptual design can be interpreted as: "*a creative problem-solving process, enabled by human knowledge, intuition, creativity and reasoning*" from a methodological point of view; "*a cognitive process, in which ideation, externalization, synthesis, and manipulation of design concepts take place in symbiosis in a short-term evolutionary process*" from a cognitive point of view; or, "*an iterative search process, in which designers gather, generate, represent, transform, manipulate, and communicate information and knowledge related to various domains of design concepts*" from an information technology point of view.

2.2.2 Importance of achieving a reliable design task

For conceptual architectural design, it is important to achieve a reliable design task. This is mainly due to the fact that achieving a reliable design task is a prerequisite to obtaining reliable design solutions (Eschenauer et al., 1992). However, a conceptual architectural design task is usually ill-defined or inaccurate. Thus, the inaccuracies of a design task should be reduced as much as possible.

2.2.3 Means to achieve a reliable design task

Design task re-definition is a potential means to achieve a more reliable design task (see FIG.2.1). For a better understanding of this, it is helpful to know its definition first, and then go into more detail about its features.

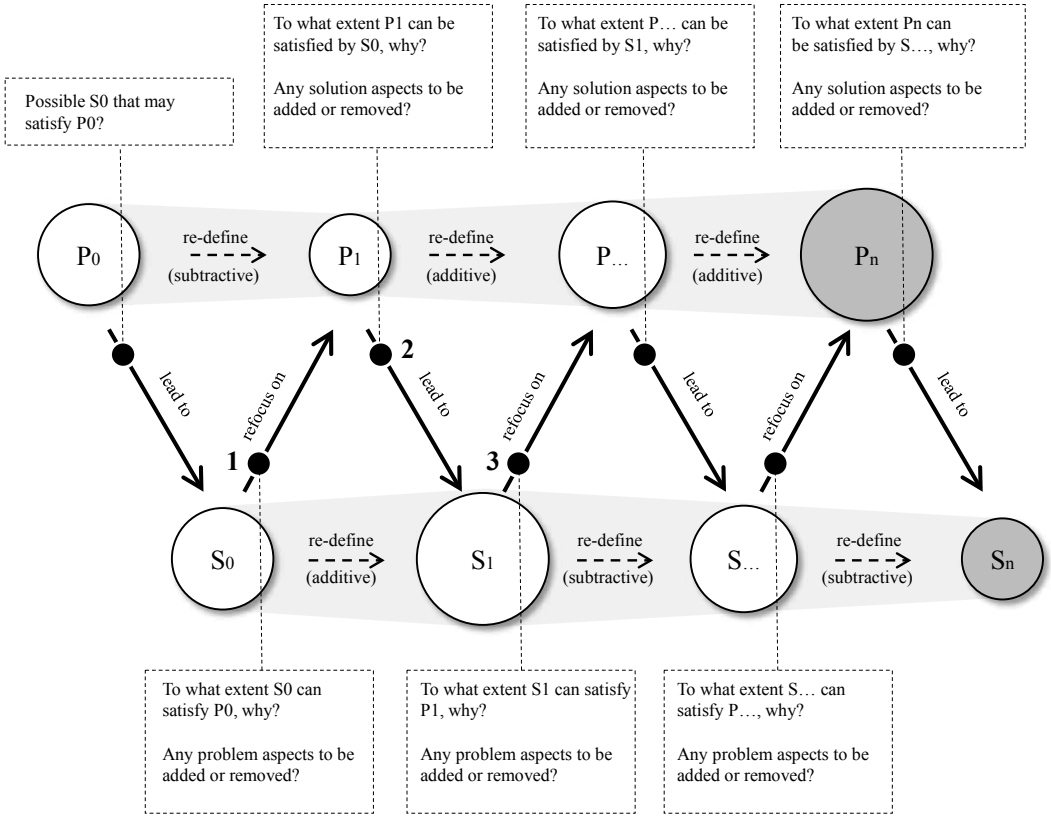


FIG. 2.1 Dynamic and interactive design task re-definition (revised from Maher et al., 1996; Maher and Poon, 1996; Dorst and Cross, 2001)

Note: P and S represent problem space (i.e., requirement space) and solution space (i.e., concept space) respectively; the dots represent knowledge extraction; and the changing sizes of the circles indicate the additive and subtractive re-definition of the space.

First, design task re-definition here refers to the re-definition of design requirements and concepts that constitute a design task. It can be considered as a design exploration (Smithers and Troxell, 1990; Navinchandra, 1991; Smithers, 1992; Jonas, 1993; Logan and Smithers, 1993; Smithers et al., 1994; Gero, 1994). More precisely, it can be seen as a phenomenon in design where problem space (i.e., requirement space) interacts and evolves with solution space (i.e., concept space) over time, namely the co-evolution of problem space and solution space (Maher et al., 1996; Maher and Poon, 1996; Dorst and Cross, 2001). Also, it can be seen as a situated process where designers interpret problems (i.e., design requirements), propose solutions (i.e., design concepts), and redefine the problems and solutions

TOC

(Hay et al, 2017). This re-definition makes it possible to shift problem space and solution space to include unexplored areas and/or exclude existing areas, thus facilitating to achieve a more reliable design task. As observed by Smith and Tjandra (1998) from their design experiments, the willingness to redefine a design task has a positive effect on the quality of a final design.

Then, design task re-definition is desired to possess the dynamic and interactive features. These features can help to leverage the full potential of the re-definition, as described in the following two sections.

2.2.3.1 Dynamic design task re-definition

Dynamic design task re-definition refers to the re-definition that can continue for multiple iterations; while in contrast, non-dynamic design task re-definition refers to the re-definition that proceeds for only one iteration. The desire for dynamic design task re-definition stems from the need to continuously extract relevant knowledge, as described below.

The necessity to continuously extract knowledge about a design task

In conceptual design, it is necessary for designers to continuously extract new knowledge. Many precedent studies pointed out that design knowledge is often insufficient in conceptual design (Paulson Jr, 1976, Fabrycky and Blanchard, 1991; Fabrycky, 1994; Mavris et al., 1998; Mavris and DeLaurentis, 2000; CURT, 2004; Blanchard and Fabrycky, 2011). In this design phase, designers may have a limited understanding on what are the proper design requirements and concepts, and how they interact with each other; furthermore, their understanding of these questions may become more limited when the design requirements and concepts can be changed dynamically.

The shortcoming of knowledge is especially true in the conceptual design of complex buildings. This type of design usually involves many possibly conflicting performance requirements and competing geometric concepts. Thus, in such a context, it is very difficult for designers to fully grasp the above-mentioned knowledge, and to get closer to the true design task.

The desire for dynamic design task re-definition

In conceptual design, dynamic design task re-definition is desired, for continuously extracting relevant knowledge. As illustrated in FIG.2.1, after defining the initial problem space (i.e., requirement space) and solution space (i.e., concept space), designers are allowed to re-define them in an iterative manner; previous spaces are continuously analyzed to acquire new knowledge, and in turn, the new knowledge is used to trigger further new re-defined spaces; such iterative re-definition process can continue, until the knowledge obtained has become insignificant, or, the improvement of designers' knowledge cannot warrant further re-definition. Thus, dynamic design task re-definition is essentially a continuous human learning process or knowledge extraction process where designers can gradually acquire new knowledge on what are the proper design requirements and concepts, and how they interact with each other.

2.2.3.2 Interactive design task re-definition

Interactive design task re-definition refers to the re-definition that can take advantage of the complementary capabilities of humans and computers; while in contrast, non-interactive design task re-definition refers to the re-definition that relies only on computer capabilities or human capabilities. The desire for interactive design task re-definition stems from the need to support quantitative and qualitative, and divergent and convergent design thinking, as described below.

The necessity to support quantitative and qualitative thinking

Both quantitative and qualitative thinking is necessary for architectural design. They can be known as *hard* and *soft,* technical and non-technical components (Sariyildiz, 2012), or rational and irrational, and objective and subjective components (Reisner-Cook, 2009) of architectural design. While there are many studies centered on quantitative aspects (e.g., those handling indoor environments, structural efficiency, energy efficiency), some other studies focus on qualitative aspects (e.g., those handling aesthetic, and formal expressions). For instance, Reisner-Cook (2009) focused on an aesthetic that he believed is still meager and simplistic; and Riccobono et al. (2013) focused on a contemporary architectural trend that is associated with formal expressions and known as Digital Expressionism.

In this research, quantitative thinking refers to the consideration of technical requirements (e.g., climatic, structural, and energy requirements), which mainly relies on computational supports; qualitative thinking refers to the consideration of non-technical requirements (e.g., aesthetic, social, and cultural requirements), which mainly relies on human subjectivity. Both kinds of thinking need to be given proper degrees of attention in conceptual design, and their relative priority may vary depending on different design contexts.

This is also true in the conceptual design of complex buildings. In this type of design, quantitative thinking (or technical rationality) is important; but it does not mean that qualitative thinking (or aesthetic preference) is not important. For example, to peruse "*green*" buildings (that have fewer impacts on the environment and cost less to maintain), designers need to highlight technical rationality while in the meantime considering aesthetic preference, that is to consider aesthetic preference based on the premise of technical rationality, rather than the other way around.

The necessity to support divergent and convergent thinking

Both divergent and convergent thinking is necessary for design, especially in conceptual design. They can be known as *divergent* and *convergent* design stages (Jones, 1992; Cross, 2008), lateral and vertical design transformations (Meniru et al., 2003), or design concept generation and selection (Pugh, 1991; Pahl et al., 2007; Ulrich and Eppinger, 2012). From the perspective of an entire design process, Jones (1992) classified a design process into three stages: divergence (i.e., a stage to extend the boundary of a design task), transformation (i.e., a stage to turn a complex design task into a simple one by deciding what to emphasize or overlook), and convergence (i.e., a stage to gradually reduce uncertainties until only one of many possible design alternatives is left as the final solution); Cross (2008) considered that the overall trend of a design process is convergent, but it still contains deliberate divergent periods to widen the search for new ideas; and Meniru et al. (2003) considered that there are two kinds of design transformations: lateral transformations (by introducing new ideas) and vertical transformations (by refining existing ideas). When particularly focusing on conceptual design, Pugh (1991) considered that conceptual design includes concept addition and reduction in an alternating way until one or a small number of final concepts are left; Pahl et al. (2007) considered that conceptual design includes concept generation, concept evaluation, and concept selection; similarly, Ulrich and Eppinger (2012) considered that conceptual design includes concept generation (i.e., a divergent process of generating alternative concepts) and concept selection (i.e., a convergent process of selecting promising concepts).

In this research, divergent thinking refers to the enrichment of new concepts (i.e., divergent concept enrichment) for which human subjectivity plays an important role; convergent thinking refers to the refinement of existing concepts (i.e., convergent concept refinement) for which computational supports are particularly helpful. Both kinds of thinking need to be given proper degrees of attention, and their relative priority may vary depending on different design contexts.

This is also true in the conceptual design of complex buildings. In this type of design, divergent thinking is often encouraged especially in the relatively early phase; while convergent thinking is conducted in the relatively late phase. Generating a wide range of concepts (i.e., divergent thinking) is helpful to prevent overlooking valuable concepts; while evaluating and selecting these concepts (i.e., convergent thinking) are useful for restricting their number from getting too large to allow meaningful considerations (Liu et al., 2003). Nevertheless, convergent thinking is sometimes prioritized. As observed by Ullman et al. (1988), Rowe (1991), Ball et al. (1994), designers may focus on one concept only; even when severe problems have been found in the concept, designers may prefer to apply patches to make the concept work rather than to reject it and develop a new one.

The desire for interactive design task re-definition

In conceptual design, interactive design task re-definition is desired, for supporting quantitative and qualitative, and divergent and convergent design thinking. As illustrated in FIG.2.1, the re-definition follows an alternating additive and subtractive process where the problem space (i.e., requirement space) and solution space (i.e., concept space) can be enlarged or shrunk. This process represents the support of quantitative and qualitative, and divergent and convergent design thinking. Both humans and computers can play an active role in supporting such design thinking. Thus, interactive design task re-definition is essentially a process to expand the variety of ways of thinking.

2.3 Performative computational design

This section provides some different interpretations of a performance approach at different times (Section 2.3.1); then, it describes the importance of achieving a reliable optimization problem in optimal design (Section 2.3.2); next, it identifies a potential means to achieve a reliable optimization problem (Section 2.3.3); and last, it preliminarily reviews optimal-design methods with a focus on dynamic and interactive Optimization Problem Re-Formulation (Re-OPF).

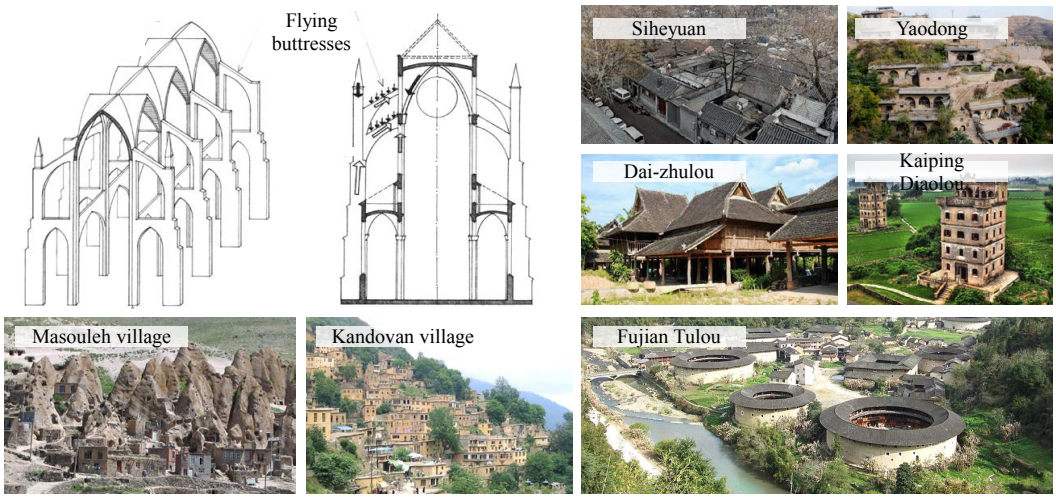
2.3.1 Interpretations of a performance approach

There is no consensus definition of a performance approach. A performance approach can be interpreted differently, as it evolves and acquires new meanings over time. Given this fact, it is worth being aware of some different interpretations.

A performance approach in ancient times

The performance approach in buildings is not new, and it can be traced back to ancient times. Early evidence of this approach includes but is not limited to the statements in ancient building regulations and architectural books and the practical examples of ancient buildings.

The performance approach has been reflected in the Code of Hammurabi and the Ten Books on Architecture (Gross, 1996). First, in the Code of Hammurabi, the performance-related statement relates to structural safety, namely: "*Article 229: If a builder builds a house for someone and does not construct it properly, and the house which he built falls in and kills its owner, then that builder shall be put to death*" (King, 2004). This statement does not prescribe the means of building, such as the thickness of walls, the sizes, and the materials of structural members. Instead, it addresses the final result, that is, the house should not collapse and kill anybody. Moreover, in the Ten Books on Architecture, the performance-related statement concerns more than just structural aspects. Vitruvius affirmed three fundamental prerequisites for a successful piece of architecture: "*firmitas, utilitas, venustas*" which are rendered in English as "*firmness, utility, beauty*" (Marconi, 2015). In both statements, user requirements for controlling the quality of buildings were given in the form of performance.



Images source:
Flying buttresses, Ching, F. D. (2011). A visual dictionary of architecture. John Wiley & Sons;
Masouleh village, https://commons.wikimedia.org/wiki/File:Masouleh.jpg;
Kandovan village, https://commons.wikimedia.org/wiki/File:Village_troglodyte_kandovan_iran.jpg;
Siheyuan, https://www.wallpaper.com/architecture/folding-courtyard-house-archstudio-china;
Dai-zhulou, https://www.re-thinkingthefuture.com/architectural-styles/a2171-vernacular-architecture-styles-of-china/;
Yaodong, https://www.sohu.com/a/280617583_100257837;
Kaiping Diaolou, https://commons.wikimedia.org/wiki/File:KaipingDiaolou.jpg;
Fujian Tulou, https://commons.wikimedia.org/wiki/File:Hekeng_-_view_from_the_lookout_-_DSCF3048.JPG.

FIG. 2.2 Performative thinking in ancient buildings

The performance approach has also been reflected in ancient buildings around the world, as shown in FIG.2.2. These buildings include not only those designed by professional architects (i.e., architect-designed buildings), but also those made by people in tribal, folk, peasant, and popular societies where an architect, or specialist designer, is not employed (i.e., vernacular buildings) (Oliver, 2006). First, one kind of example that reflect performative thinking is the "*flying buttresses*" of Gothic churches. The form of flying buttresses was a result of performative considerations, including structural and spiritual considerations (Kanaani, 2015). Specifically, the form was designed to carry the lateral forces of the vault away from the upper walls towards the external vertical columns, and meanwhile to admit more daylight and create slender facade elements (because visual impressions of lightness and verticality are deemed respect for heaven). Moreover, other examples that reflect performative thinking are vernacular dwellings. The forms of the dwellings are usually performative responses to their local conditions, namely local climate, materials, tradition, etc. (Kanaani, 2015). Specifically, some of the dwellings respond to their climatic conditions, such as mud-caved houses "*Yaodong*", courtyard houses

TOC

"*Siheyuan*", stilt houses "*Dai-zhulou*" in different climate zones of China; earth houses in Masouleh village, rock-caved houses in Kandovan village, wind-catchers in the city of Yazd in Iran. Some of the dwellings respond to their historical conditions, such as earth houses "*Fujian Tulou*" and tower houses "*Kaiping Diaolou*" in China, which were built by prioritizing defense requirements.

A performance approach in modern times

The performance approach in buildings has been formally developed since the early 20th century. The development of this approach has provided an important theoretical foundation for its further development.

An influential definition of the performance approach is provided by the CIB Working Commission W60, namely: "*the performance approach is, first and foremost, the practice of thinking and working in terms of ends rather than means. It is concerned with what a building or building product is required to do, and not with prescribing how it is to be constructed*" (Gibson, 1982). This approach can be applied in the forms of performance-based building codes and performance-based building design. Many countries worldwide have committed to developing performance-based building codes during the modern development period (Gross, 1996; Szigeti and Davis, 2005). According to the U.S. Building Code Committee (1925): "*whenever possible, requirements should be stated in terms of performance … rather than in dimensions, detailed methods, or specific materials*". The development of performance-based building codes can promote the application of performance-based building design, such as performance-based seismic design, or fire safety design.

The performance approach can overcome a major drawback of a traditional prescriptive approach. More precisely, it can be used to generate more innovative designs, at the cost of increasing implementation complexity. In the prescriptive approach, proposed designs just need to comply with prescriptive requirements or codes which describe the "*means*", and no performance assessment is needed. This can make the prescriptive approach relatively simpler to implement (Becker, 2008) but result in a major drawback - stifling design innovation (Szigeti and Davis, 2005). Differently, in the performance approach, proposed designs need to meet performance requirements or codes which describe the "*ends*", and a performance assessment is required. This can result in more innovative designs (Hien et al., 2000) but make the performance approach relatively more complex to implement (Becker, 2008). In addition, the performance approach has other advantages as well, such as facilitating the satisfaction of user needs, facilitating communication among stakeholders, and facilitating international trade in building products. (Hattis and Becker, 2001).

The performance approach is especially meaningful for complex projects than for simple projects, given the following facts. For simple projects (e.g., those with regular shapes and common size elements), there are usually many well-proven technologies available; thus, it is possible to use the prescriptive approach that is faster, less costly, and more reliable for ensuring acceptable levels of performance (Becker, 2008). For complex projects (e.g., those with irregular shapes and oversize elements), innovations and optimal solutions are usually their main concerns; thus, it is indispensable to use the performance approach that allows to test of more innovative options and has more chances to reach optimal levels of performance (Becker, 2008). In fact, it is not likely to use the performance approach or the prescriptive approach individually in many cases. In other words, the use of the performance approach does not preclude the use of the prescriptive approach; it is often better to blend these two approaches, given their complementary capabilities (Szigeti and Davis, 2005; Huovila et al., 2018).

A performance approach in recent times

The performance approach in buildings has received more attention with the rising popularity of computational methods in recent decades, thus it is also known as performance-based computational design. A recent trend is to apply this approach to conceptual architectural design.

The performance approach can be defined differently by highlighting different aspects of computational or digital support. For instance, it can be defined as: "*a digital architectural design approach in which broadly understood building performance is a guiding design principle on a par with or above form-making,*" and termed "*performative architecture*" (Kolarevic, 2003a; Kolarevic, 2004; Kolarevic, 2005). It aims to shift the emphasis: from form-making to form-finding (Kolarevic, 2003b); or, from an architecture purely based on visual concerns to an architecture justified by its performance (Leach, 2009). Moreover, Malkawi (2005) considered that the performance approach can be supported by digital simulation tools, optimization, and partial automation, which is termed "*performance-driven design*." It aims to shift the conventional use of simulation tools from analysis only to analysis and synthesis (Malkawi, 2005). Oxman (2008a; 2008b) considered that the performance approach can be supported by evaluative simulation, digital form generation, and modification, which is termed "*performative design*." It aims to shift from a design paradigm externally controlled by human designer's formal manipulative skills to that internally informed by computer's evaluative and simulation processes (Oxman, 2008a; Oxman, 2008b).

The performance approach has been increasingly applied to the conceptual design of contemporary architecture. It has become a prevailing paradigm; many designs associated with digital architecture can reflect, or partially reflect, its application (Oxman et al, 2007). Some famous example projects that apply this approach are shown in FIG.2.3. For the Swiss RE Building (2004), the curved overall shape and the façade structural framing were designed based on wind performance simulations, to reduce the wind impact on the building's perimeter and the nearby street. For the Greater London Authority Headquarters (2002), the inclined overall shape was designed based on energy performance simulation; that is, the building surface exposed to direct sunlight is minimized, to reduce energy use. For the Kunsthaus Graz (2003) and the Beijing Olympic Stadium (2008), the overall shape and the façade structural framing were adjusted based on structural performance simulation, to improve the structural performance.

Images source:
Swiss RE Building, https://commons.wikimedia.org/wiki/File:30_St_Mary_Axe_from_Leadenhall_Street.jpg;
Greater London Authority Headquarters, https://commons.wikimedia.org/wiki/File:London_City_Hall.jpg;
Kunsthaus Graz, https://commons.wikimedia.org/wiki/File:Graz_Kunsthaus_vom_Schlossberg_20061126.jpg
Beijing Olympic Stadium, https://commons.wikimedia.org/wiki/File:Beijing_national_stadium.jpg.

FIG. 2.3 Example projects that apply performance-based computational design

2.3.2 Importance of achieving a reliable optimization problem

For optimal design, it is important to achieve a reliable optimization problem. This is mainly because achieving a reliable optimization problem is a prerequisite to obtaining reliable optimal solutions. However, unavoidable inaccuracies or simplifications occur when constructing an optimization problem for a design task (Meignan et al., 2015). Thus, the inaccuracies of an optimization problem should be reduced as much as possible.

2.3.3 Means to achieve a reliable optimization problem

Optimization Problem Re-Formulation (Re-OPF) is a potential means to achieve a more reliable optimization problem. For a better understanding of this, it is helpful to know its definition first, and then go into more detail regarding its features.

First, Optimization Problem Re-Formulation (Re-OPF) here refers to the re-formulation of objectives, constraints, and design variables that constitute an optimization problem. It can be seen as a response to design task re-definition (mentioned in Section 2.2.3). More precisely, once design requirements and concepts of a design task evolve during design exploration, the objectives and constraints used to describe the design requirements, and the design variables used to define the design concepts need to be adapted accordingly. This re-formulation makes it possible to shift objective space and design space to include unexplored areas and/or exclude existing areas, thus facilitating to achieve a more reliable optimization problem. As stated by Arora (2016), re-formulating an optimization problem can help to avoid unrealistic solutions or impractical designs.

Then, Optimization Problem Re-Formulation (Re-OPF) is desired to possess dynamic and interactive features. These features can help to leverage the full potential of the re-formulation, as described in the following two sections.

2.3.3.1 Dynamic optimization problem re-formulation

Dynamic Optimization Problem Re-Formulation (Re-OPF) refers to the re-formulation that can continue for multiple iterations; while in contrast, non-dynamic Optimization Problem Re-Formulation (Re-OPF) refers to the re-formulation that proceeds for only one iteration. The desire for dynamic Optimization Problem Re-Formulation (Re-OPF) stems from the need to continuously extract relevant knowledge, as described below.

The necessity to continuously extract knowledge about an optimization problem

In optimal design, it is necessary for designers to continuously extract new knowledge. Meignan et al. (2015) stated that an optimization problem is actually an approximation to a design task, namely that there is usually a discrepancy between an optimization problem and a design task. This discrepancy indicates the insufficiency of relevant knowledge. It is often the case that designers may

have a limited understanding on what are the proper performance objectives and constraints to describe design requirements, and what are the proper design variables to define design concepts; furthermore, their understanding of these questions may become even more limited, when the objectives, constraints, and design variables are going to change dynamically.

The shortcoming of knowledge is especially true in the optimal design of complex buildings. This type of design usually involves many possible choices of objectives, constraints, and design variables. Thus, in such a context, it is very difficult for designers to fully grasp the above-mentioned knowledge in one shot, and to correctly convert a design task into an optimization problem.

The desire for dynamic optimization problem re-formulation

In optimal design, dynamic Optimization Problem Re-Formulation (Re-OPF) is desired, for continuously extracting relevant knowledge. Such re-formulation follows a similar iterative process to dynamic design task re-definition (mentioned in Section 2.2.3.1). Thus, it is essentially a continuous human learning process or knowledge extraction process where designers can gradually improve their understanding of the proper performance objectives and constraints to describe design requirements, the proper design variables to define design concepts, and how the performance objectives and constraints interact with the design variables.

2.3.3.2 Interactive optimization problem re-formulation

Interactive Optimization Problem Re-Formulation (Re-OPF) refers to the re-formulation that can take advantage of the complementary capabilities of humans and computers; while in contrast, non-interactive Optimization Problem Re-Formulation (Re-OPF) refers to the re-formulation that relies only on computer capabilities or human capabilities. The desire for interactive Optimization Problem Re-Formulation (Re-OPF) stems from the need to support quantitative and qualitative performance measures, and divergent and convergent design variables, as described below.

The necessity to support quantitative and qualitative performance measures

Both quantitative and qualitative performance measures are necessary for optimal design. They can be seen as a response to supporting quantitative and qualitative thinking (mentioned in Section 2.2.3.2). In this research, quantitative performance measures refer to those used to assess the achievement of technical requirements, while qualitative performance measures refer to those used to assess the achievement of non-technical requirements.

The necessity to support divergent and convergent design variables

Both divergent and convergent design variables are necessary for optimal design. They can be seen as a response to supporting divergent and convergent thinking (mentioned in Section 2.2.3.2). In this research, divergent design variables refer to those added to consideration for enriching new concepts or those whose domains are expanded, while convergent design variables refer to those removed from consideration for refining existing concepts or those whose domains are shrunk.

The desire for interactive optimization problem re-formulation

In optimal design, interactive Optimization Problem Re-Formulation (Re-OPF) is desired, for supporting quantitative and qualitative performance measures, and divergent and convergent design variables. The re-formulation also follows an alternating additive and subtractive process (similar to the interactive design task re-definition mentioned in Section 2.2.3.2). This process represents the support of quantitative and qualitative performance measures and divergent and convergent design variables. Both humans and computers can play an active role in supporting those performance measures and design variables. Thus, interactive Optimization Problem Re-Formulation (Re-OPF) is essentially a process to expand the variety of performance measures and design variables.

2.3.4 A preliminary review of optimal-design methods

A preliminary review of optimal-design methods is conducted in this section, to understand the general state of the art in supporting Optimization Problem Re-Formulation (Re-OPF), especially dynamic and interactive re-formulation.

Optimal-design methods are highly useful for building design. They are not new to building engineering disciplines, because they have been applied in these disciplines for a long time (Arora, 1990). With the rapid progress of computer science in recent decades, they have been increasingly applied to different building disciplines and design phases (Evins, 2013; Nguyen et al., 2014; Huang and Niu, 2016), including conceptual architectural design (Østergård et al., 2016; Touloupaki and Theodosiou; 2017; Ekici et al., 2019).

Current optimal-design methods have rarely considered Optimization Problem Re-Formulation (Re-OPF). As stated by Bernal et al. (2015), designers can rapidly identify relevant aspects of a design task and constantly shift the direction of design development through re-definition but receive little or no computational support for such behavior. For instance, there is often a lack of Optimization Problem Re-Formulation (Re-OPF). This is usually associated with an implicit and incorrect assumption, namely, an ill-defined conceptual architectural design task is incorrectly assumed as a well-defined design task where all objectives, constraints, and design variables are not replaceable and removable. Given this assumption, designers may just focus on searching for optimal solutions based on a fixed and probably premature optimization problem (due to the lack of discussing the reliability of the optimization problem). The absence of Optimization Problem Re-Formulation (Re-OPF) hinders designers from thinking outside of the box and expanding the exploration into vast possible regions beyond the original space. This is not beneficial for achieving a more reliable optimization problem and avoiding unrealistic solutions or impractical designs.

It is even rarer that current optimal-design methods have considered dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). Related gaps can be described below.

Lack of considering continuous knowledge extraction

Despite the necessity of continuously extracting relevant knowledge, current optimal-design methods have just begun to notice it. It was not until recently that the value of dynamic re-formulation in knowledge extraction was paid more attention to. As stated by Arora (2016), the process of developing a proper formulation for the optimal design of a practical problem is iterative in itself; in this iterative process, new knowledge can be extracted continuously to support the revision of the initial formulation. Nevertheless, this iterative process has not been widely considered in current optimal-design methods. This reflects the lack of dynamic re-formulation.

Lack of considering qualitative performance measures

Despite the necessity of supporting quantitative and qualitative performance measures, current optimal-design methods have rarely considered qualitative performance measures. As stated by Østergård et al. (2016), current optimal-design methods have an important drawback – the lack of qualitative measures which are critical for conceptual design. This can be associated with the fact that human subjectivity (e.g., human preferences, intuitions, and emotions that are important for supporting qualitative measures) is often excluded in optimal-design methods. The absence of human subjectivity is not beneficial for assessing non-technical performances; it results in that conceptual architectural design is erroneously treated as a process where qualitative matters are not considered as a necessity. This reflects the lack of iterative re-formulation.

Lack of considering divergent design variables

Despite the necessity of supporting divergent and convergent design variables, current optimal-design methods have rarely considered divergent design variables for supporting a divergent design process. As stated by Bernal et al. (2015), current optimal-design methods have a major challenge – supporting a divergent conceptual design process instead of a convergent one. This can be associated with the fact that human creativity (i.e., a divergent thinking style that can lead to creativity) is often ignored in optimal-design methods. The absence of human creativity is not beneficial for expanding the domains of design variables and adding new design variables; it results in that conceptual architectural design is erroneously treated as a process where divergent thinking is not considered as a necessity. This reflects the lack of iterative re-formulation.

2.4 Review of multi-objective optimization design methods

This section reviews relevant Multi-Objective Optimization (MOO) design methods. First, it specifies the scope of the review (Section 2.4.1); then, it presents the methods that are classified into four types (Section 2.4.2); and finally, it identifies the gaps of the methods (Section 2.4.3). This review has been updated as of April 2020.

2.4.1 Scope of the review

Multi-Objective Optimization (MOO) design methods are the subject of this review. Recently, there are a growing number of Multi-Objective Optimization (MOO) design methods applied to conceptual architectural design (Østergård et al., 2016; Touloupaki and Theodosiou; 2017; Ekici et al., 2019). They adopt the following techniques to varying extents: parametric geometric modeling (Aish and Woodbury, 2005) and sampling algorithms for geometry generation; multi-disciplinary simulation modeling (Hensen and Lamberts, 2011; Gaetani et al., 2020) for performance analysis; Multi-Objective Optimization (MOO) algorithms (Andersson, 2000; Deb, 2014) for optimization; quantitative data analysis and qualitative data visualization for information and knowledge extraction. Among these techniques, data analysis is particularly important for supporting Optimization Problem Re-Formulation (Re-OPF). Here, data analysis refers to the process of studying a given data set in close detail to extract useful information; it differs from data analytics which is a more comprehensive term referring to a discipline that comprises the complete management of data, including collection, cleaning, organizing, storing, administering, and analysis of data (Sarangam, 2020).

This review covers Multi-Objective Optimization (MOO) design methods that can support Optimization Problem Re-Formulation (Re-OPF) in different ways, given the value of such re-formulation. The reviewed methods are presented in Appendix I and described below. Note that they are not limited to those developed for conceptual architectural design, due to the lack of relevant studies.

2.4.2 Types of multi-objective optimization design methods

According to whether or not Optimization Problem Re-Formulation (Re-OPF) is dynamic and interactive, the reviewed Multi-Objective Optimization (MOO) design methods are classified into four types (see FIG.2.4). This classification can help to understand the state of the art of methods in supporting Optimization Problem Re-Formulation (Re-OPF).

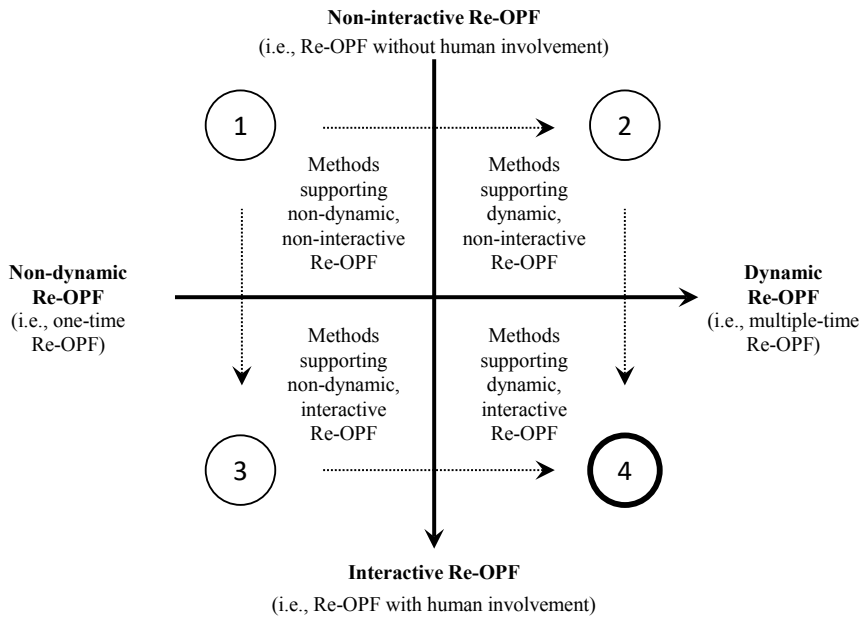


FIG. 2.4 Classification of the reviewed Multi-Objective Optimization (MOO) design methods (Yang et al., 2020)

2.4.2.1 Type 1 methods incorporating non-dynamic and non-interactive Re-OPF

Type 1 methods incorporate non-dynamic and non-interactive re-formulation of an optimization problem (i.e., one-time re-formulation where qualitative objectives and divergent design variables are not considered). The necessity of allowing such re-formulation has been pointed out in the aerospace and automotive industries for nearly a decade. Agte et al. (2010) stated that it is necessary to include design space re-definition in optimization, given the fact that design requirements may

TOC

change over time and significant re-designs can occur at a later time. Similarly, Simpson and Martins (2011) considered that allowing the change of design variable sets can help to explore new regions of the design space and lead to better designs.

Regarding Type 1 methods, there are a few examples in the late phases of architectural design. Heiselberg et al. (2009)'s example, Shen and Tzempelikos (2013)'s example are two typical ones. In these examples, the original design space is refined based on sensitivity analysis results. That is, sensitivity analysis of multiple performance metrics to various design variables is conducted; then, the design variables are ranked according to their relative importance to each of the performance metrics. In this way, unimportant design variables are identified and screened out, thus refining the original design space. Given that design variables are removed once, qualitative objectives and divergent design variables are not considered, the above examples belong to Type 1 methods.

2.4.2.2 Type 2 methods incorporating dynamic and non-interactive Re-OPF

Type 2 methods incorporate dynamic and non-interactive re-formulation of an optimization problem (i.e., multiple-time re-formulation where qualitative objectives and divergent design variables are not considered). The necessity of allowing such re-formulation has been noticed recently in the building industry. Arora (2016) stated that developing a proper formulation for a design optimization problem is an iterative process, namely the initial formulation often needs several adjustments before obtaining an acceptable one.

Regarding Type 2 methods, there are some examples but not in conceptual architectural design. These example methods, known as "dynamic Multi-Objective Optimization (MOO)," are a hot research topic in computer science (Raquel and Yao, 2013; Helbig and Engelbrecht, 2013; Azzouz et al., 2017). They are meant for solving Multi-Objective Optimization (MOO) problems which involve time-varying objectives, constraints and design variables, such as control problems, scheduling problems, and mechanical design problems. (Helbig and Engelbrecht, 2014). To solve these problems, the Multi-Objective Optimization (MOO) algorithms used should be able to track the changing Pareto-optimal fronts. These algorithms include at least two kinds: algorithms which solve a dynamic optimization problem without adapting the problems, as applied in Trabelsi et al. (2016)'s example; and algorithms which convert a dynamic optimization problem into multiple static optimization problems, as applied in Curtis et al. (2013)'s two examples. The latter

kind of algorithms indicates that a static optimization problem can be re-formulated iteratively by adding and removing objectives and design variables, and that the re-formulation can extend the exploration into a much larger space and avoid missing potentially better solutions. Given that quantitative objectives and/or design variables are re-formulated multiple times, qualitative objectives and divergent design variables are not considered, the above examples belong to Type 2 methods.

2.4.2.3 Type 3 methods incorporating non-dynamic and interactive Re-OPF

Type 3 methods incorporate non-dynamic and interactive re-formulation of an optimization problem (i.e., one-time re-formulation where qualitative objectives and/or divergent design variables are considered). The necessity of allowing such re-formulation has been noticed in the building industry. Cichocka et al. (2017)'s survey among architects showed that 91% of the surveyed architects would like to influence optimization outcomes in a subjective way like subjectively selecting promising designs. This indicates that human-in-the-loop methods seem more appropriate in architectural design optimization. Brintrup et al. (2007) deemed that an optimization framework should be flexible enough to deal with changeable (qualitative and quantitative) objectives, constraints, and preferences. Mueller and Ochsendorf (2015) stated that an ideal computational approach should expose designers to a diverse range of alternatives which may inspire new goals and spark new ideas.

Regarding Type 3 methods, there are some examples in conceptual architectural design and other design disciplines. These example methods are known as "interactive evolutionary computation" which belongs to human-in-the-loop optimization methods. They often adopt human subjectivity to evaluate hard-to-quantify qualitative performances (Takagi, 2001), such as the value or beauty of buildings that can be quickly captured through human observations (Graf, 1995; Graf 1996). They are versatile in handling changing definitions of qualitative objectives, because hard-coding qualitative influences is not necessary (Brintrup et al., 2007). The examples from Brintrup et al. (2007), Mueller and Ochsendorf (2015), and Turrin et al. (2011) are typical interactive evolutionary computation applied in conceptual architectural and structural design. In these examples, first, a single-objective optimization involving one quantitative objective is run, thus obtaining quantitative promising designs; then, human designers are asked to evaluate the qualitative performance of the obtained designs subjectively and select preferred ones for further optimization. In a situation that there is no further optimization, the original optimization problem can be seen as being re-

formulated once by removing a quantitative objective and adding a qualitative objective. Differently, Barnum and Mattson (2010)'s example is a reverse method. In this example, first, a single-objective optimization involving one qualitative objective is run based on human subjective evaluation, thus obtaining a quantitative preference-based model; then, the original optimization problem is re-formulated by adding quantitative objectives; last, the re-formulated problem is run based on the preference-based model and other related physics-based models. Given that objectives are re-formulated once, qualitative objectives are considered, the above examples belong to Type 3 methods.

2.4.2.4 Type 4 methods incorporating dynamic and interactive Re-OPF

Type 4 methods incorporate dynamic and interactive re-formulation of an optimization problem (i.e., multiple-time re-formulation where qualitative objectives and/or divergent design variables are considered). The necessity of allowing such re-formulation is rarely noticed in the building industry. Newton (2018) pointed out three limitations of Multi-Objective Optimization (MOO) design methods used in architectural design, namely, they are not designed for finding novel and diverse designs; they are not suitable for open-ended iterative design processes where design space and objective space may dynamically change; and they cannot bring designers into the loop in ways that stimulate the designers to be more creative. These limitations actually indicate the necessity of dynamic and interactive re-formulation. Janssen (2015) suggested an adaptive-iterative design process that allows designers to re-define a design space dynamically and interactively.

Regarding Type 4 methods, there are very few examples in conceptual architectural design. Newton (2018)'s example, Kaushik and Janssen (2013)'s example are two valuable ones. In these two examples, it is through dynamic and interactive re-formulation that the design processes are driven forward, and that the designs are made more complex and less abstract progressively; but such re-formulation is realized in different ways. In the former example, human designers engage in re-defining qualitative objectives and devising divergent design variables; quantitative and qualitative objectives are added and/or removed two times; divergent design variables are added two times. In the latter example, human designers engage in devising divergent design variables but not in re-defining qualitative objectives; quantitative objectives are added once; divergent design variables are added and/or removed three times. Given that objectives and/or design variables are re-formulated multiple times, qualitative objectives and/or divergent design variables are considered, the above examples belong to Type 4 methods.

2.4.3 Gaps of multi-objective optimization design methods

According to the above review, there are a very small number of Multi-Objective Optimization (MOO) design methods that have incorporated dynamic and interactive Optimization Problem Re-Formulation (Re-OPF), in conceptual architectural design. Specifically, most of the reviewed methods have not incorporated such re-formulation (i.e., Type 1, Type 2 and Type 3 methods); while in contrast, only two of the reviewed methods have done so (i.e., Type 4 methods). However, even in the promising Type 4 methods (e.g., Newton's, Kaushik and Janssen's design method), there is still room for improvement, especially in terms of information and knowledge extraction.

A possible reason for the above problem relates to an implicit and incorrect assumption, that is, an ill-defined task in conceptual architectural design has often been incorrectly assumed as a well-defined task. This assumption indicates that all given design requirements and concepts, and all given objectives, constraints and design variables are not replaceable and removable. Under this assumption, designers usually go quickly through the Optimization Problem Formulation (OPF) phase without thoroughly discussing the reliability of the given optimization problem, and then just focus on searching for optimal solutions based on the fixed and probably premature optimization problem in the Optimization Problem Solving (OPS) phase. Thus, this assumption can lead to a high risk of obtaining meaningless design solutions.

Another possible reason can be the insufficient awareness on the potentials of information and knowledge extraction for optimal-design methods. In most of the previous optimal-design methods for conceptual architectural design, information and knowledge extraction is at most treated as a side topic. In this situation, designers may just perform simple data analysis by using limited statistical techniques (e.g., descriptive statistics) based only on optimal solutions, thus obtain limited knowledge. But, in fact, they could have performed more sophisticated data analysis by using advanced statistical techniques (i.e., beyond descriptive statistics) based on both optimal and non-optimal solutions (Turrin et al., 2011), thus obtain more knowledge.

In summary, it is necessary to establish a new Multi-Objective Optimization (MOO), especially Multi-Objective and Multi-Disciplinary Optimization (MOMDO), design method that can incorporate dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) in a better manner. The method development will be elaborated in Chapter 3.

2.5 Review of multi-objective optimization software workflows

This section reviews relevant Multi-Objective Optimization (MOO) software workflows. First, it specifies the scope of the review (Section 2.5.1); then, it presents the software workflows that are classified into four types (Section 2.5.2); and finally, it identifies the gaps of the software workflows (Section 2.5.3). This review has been updated as of April 2020.

2.5.1 Scope of the review

Multi-Objective Optimization (MOO) software workflows are the subject of this review. Currently, there are a growing number of Multi-Objective Optimization (MOO) software workflows available for conceptual architectural design. They can be developed by integrating different software using different integration approaches. Among possible options, Visual Programming (VP) software (Boshernitsan and Downes, 2004) and Process Integration and Design Optimization (PIDO) software (Flager et al., 2009a) are desired to be integrated, given their capabilities in offering computational techniques adopted by Multi-Objective Optimization (MOO) design methods. Here, Visual Programming (VP) software refers to a particular type of parametric geometric modeling software for architects who may not have knowledge of textual programming; and it has potential to offer user-friendly parametric geometric modeling and various types of multi-disciplinary simulation modeling. Moreover, Process Integration and Design Optimization (PIDO) software refers to a particular type of optimization software that was originally used in the aerospace industry and later tested in the building industry; and it has potential to offer various types of multi-disciplinary simulation modeling, Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, and qualitative data visualization.

This review covers Multi-Objective Optimization (MOO) software workflows that can integrate Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software to varying extents, given the value of such software. The reviewed workflows are presented in Appendix II and described below. Note that they include those developed for conceptual architectural design but exclude those focusing on non-geometric parametric modeling and mono-

disciplinary simulation modeling, such as MultiOpt (Chantrelle et al., 2011), IDA-ICE+MATLAB (Hamdy et al., 2011), jEPlus+EA (Porritt et al., 2012), MOBO (Palonen et al., 2013).

2.5.2 Types of multi-objective optimization software workflows

According to whether or not Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software are integrated, the reviewed Multi-Objective Optimization (MOO) software workflows are classified into four types (see FIG.2.5). This classification can help to understand the state of the art of the workflows in utilizing Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software.

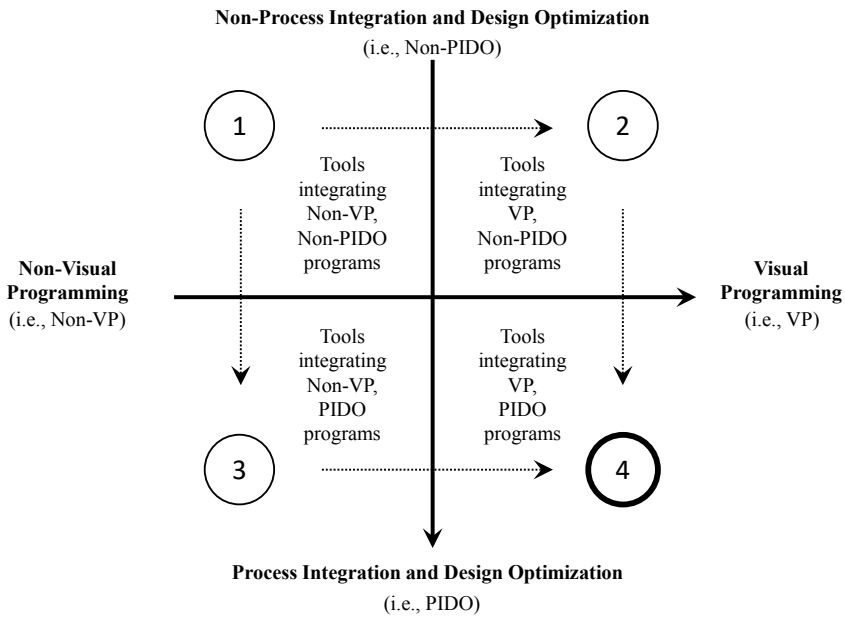


FIG. 2.5 Classification of the reviewed Multi-Objective Optimization (MOO) software workflows

TOC

2.5.2.1 Type 1 workflows integrating Non-VP and Non-PIDO software

Type 1 workflows integrate Non-Visual Programming (Non-VP) software and Non-Process Integration and Design Optimization (Non-PIDO) software. The integration approaches used include a model-based approach or a BIM file-based approach (Bernal et al., 2015). This type of software workflows generally offers less user-friendly parametric geometric modeling, and less types of multi-disciplinary simulation modeling, Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, and less ideal qualitative data visualization, as exemplified below.

Caldas (2001, 2006, 2008), Caldas and Norford (2002, 2003), and Wright et al. (2014)'s software workflows are typical examples of this type. Each of them offers textual programming (i.e., less user-friendly parametric geometric modeling), two types of simulation modeling, one type of Multi-Objective Optimization (MOO) algorithm, a simple sampling algorithm (i.e., random sampling), one type of quantitative data analysis (i.e., trade-off analysis), and separated data visualization (i.e., less ideal visualization that shows building geometries and simulation results separately).

Shea et al. (2006), Conti (2013), and Conti et al. (2015)'s software workflows are similar to the typical ones, except that each of them offers combined data visualization (i.e., more ideal visualization that shows building geometries and simulation results side-by-side simultaneously).

Gagne and Andersen (2010), Gerber and Lin (2012, 2014)'s software workflows are similar to the typical ones, except that each of them offers BIM modeling (i.e., a geometric modeling technique with limited parametric capabilities).

DesignBuilder software is also similar to the typical ones, except that it offers fast modeling (i.e., a geometric modeling technique with limited parametric capabilities), and a few more types of simulation modeling, sampling algorithms, and quantitative data analysis.

2.5.2.2 Type 2 workflows integrating VP and Non-PIDO software

Type 2 workflows integrate Visual Programming (VP) software and non-Process Integration and Design Optimization (Non-PIDO) software. The integration approach used is a custom system-to-system approach (Bernal et al., 2015). This type of software workflows generally offers more user-friendly parametric geometric modeling, more types of multi-disciplinary simulation modeling, less types of Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, and more ideal qualitative data visualization, as exemplified below.

Janssen et al. (2011), Janssen (2013, 2015), Von Buelow (2012, 2016), Vierlinger and Bollinger (2014), Negendahl and Nielsen (2015), Danhaive and Mueller (2015), Brown and Mueller (2016), and Brown et al. (2016)'s software workflows are similar examples of this type. Each of them offers visual programming, a few more types of simulation modeling, one or a few types of Multi-Objective Optimization (MOO) algorithms, a simple sampling algorithm, one or a few types of quantitative data analysis, and separated or combined data visualization.

2.5.2.3 Type 3 workflows integrating non-VP and PIDO software

Type 3 workflows integrate Non-Visual Programming (Non-VP) software and Process Integration and Design Optimization (PIDO) software. The integration approach used is a BIM file-based approach (Bernal et al., 2015). This type of software workflows generally offers less user-friendly parametric geometric modeling, many types of multi-disciplinary simulation modeling, Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, and more ideal qualitative data visualization, as exemplified below.

Flager et al. (2009b)'s software workflow is a typical example of this type. It offers BIM modeling, broad types of simulation modeling, Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, combined data visualization.

2.5.2.4 Type 4 workflows integrating VP and PIDO software

Type 4 workflows integrate Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software. The integration approach used is a custom system-to-system approach (Bernal et al., 2015). This type of software workflows generally offers more user-friendly parametric geometric modeling, many types of multi-disciplinary simulation modeling, Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, and more ideal qualitative data visualization, as exemplified below.

ESTECO's earliest in-house workflow that integrates Grasshopper and modeFRONTIER is a typical example of this type. It offers visual programming, broad types of simulation modeling, Multi-Objective Optimization (MOO) algorithms, sampling algorithms, quantitative data analysis, combined data visualization.

2.5.3 Gaps of multi-objective optimization software workflows

According to the above review, there are a very small number of Multi-Objective Optimization (MOO) software workflows that have integrated Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software, in conceptual architectural design. Specifically, most of the reviewed workflows have not integrated such software (i.e., Type 1, Type 2 and Type 3 workflows); while in contrast, only one of the reviewed workflows have done so (i.e., Type 4 workflow). However, even in the promising Type 4 workflow (i.e., ESTECO's earliest in-house workflow that integrates Grasshopper and modeFRONTIER), there is still room for improvement, especially in terms of software integration (i.e., the integration of Grasshopper and modeFRONTIER).

In summary, it is necessary to establish an improved Multi-Objective Optimization (MOO), especially Multi-Objective and Multi-Disciplinary Optimization (MOMDO), software workflow where Grasshopper and modeFRONTIER are integrated in a better manner. The workflow development will be elaborated in Chapter 4.

2.6 Review of an application field – sports buildings

This section reviews a valuable application field: the conceptual design of sports buildings, especially indoor sports halls. First, it specifies the scope of the review (Section 2.6.1); then, it identifies the gaps of conceptual sports building design (Section 2.6.2).

2.6.1 Scope of the review

Multi-Objective Optimization (MOO) applications to the conceptual design of indoor sports halls are the main subject of this review. Indoor sports halls are a particular type of complex buildings or more specifically large-span buildings that hold dry indoor sports (e.g., basketball, volleyball, five-a-side soccer, badminton, and martial arts) or wet indoor sports (e.g., swimming, and diving) (John and Heard, 1981). A middle or large scaled sports hall often consists of a competition venue, a training venue, spectator grandstands, and other auxiliary space; while, a small scaled sports hall often only includes a training venue without spectator grandstands, and a small amount of auxiliary space. The conceptual design of indoor sports halls is multi-disciplinary and multi-objective by nature; and it is complex in terms of geometries and performances. Thus, the conceptual design of indoor sports halls is a valuable field to which Multi-Objective Optimization (MOO) design methods and software workflows are applied.

This review is not limited to the above-mentioned field, due to the lack of relevant studies. More precisely, this review covers the conceptual sports building design that applies Multi-Objective Optimization (MOO), Single-Objective Optimization (SOO), or no optimization methods. The reviewed studies are presented in Appendix III and described below.

2.6.2 Gaps of conceptual sports building design

The reviewed studies can be classified in different ways, such as by whether or not optimal-design methods are applied, by building geometries and/or by building performances. Via the analysis of the studies based on different classifications, relevant gaps of conceptual sports building design can be found.

2.6.2.1 Lack of applying optimal-design methods

Optimal-design methods are highly useful for conceptual architectural design (mentioned in Section 2.3.4). This is also true when it comes to conceptual sports building design. Optimization involves formalizing design tasks so that iterative computation, both interactive and automated, can be used to find feasible and performance-driven design solutions that would be difficult to arrive at using only conventional computing and design processes (Culley and Pascoe, 2009). It allows more rigorous early exploration, evaluation, and analysis; and extracting more information from a wider range of possible solutions allows designers to explore more innovative ideas and develop better design solutions (Thornton Tomasetti, 2017).

Despite the usefulness of optimal-design methods, current conceptual design processes of sports buildings have not taken full advantages of them. Current conceptual design processes of sports buildings generally rely only on designers' experience or the knowledge derived from relevant studies (Sun et al., 2013; Zhao and Mei, 2013; Rajagopalan and Luther, 2013; Joseph et al., 2015; Suo et al., 2015; Nord et al., 2015; Cheng and Bahnfleth, 2016; Ding, 2017; Heinzelmann, 2018; Josa et al. 2020). These studies quantitatively explore the impacts of selected variables or design strategies on particular performances, but do not necessarily focus on the feedback loops from performance assessment to design modification. They may have different focuses. For instance, Zhao and Mei (2013) focus on assessing the relative impacts of different design factors on interior daylighting and formulating design recommendations; Josa et al. (2020) focus on analyzing different aspects of sustainability of structural components by means of a multi-criteria decision-making method (i.e., a weighted sum method). In these studies, optimization algorithms are not employed; instead, statistical techniques like random samplings are often used.

In recent years, discussions about optimal-design methods for conceptual sports building design have increased. Some studies utilize Single-Objective Optimization (SOO) design methods (Arkinstall and Carfrae, 2006; Holzer et al., 2007; Flager et al., 2009b; Shi and Yang, 2013; Zargar and Alaghmandan, 2019; Bianconi et al., 2020), while just a few studies utilize Multi-Objective Optimization (MOO) design methods (Brown and Mueller, 2016; Yang et al., 2018; Pan et al., 2019). Thus, it is worth studying the application of Multi-Objective Optimization (MOO) design methods in conceptual sports building design.

2.6.2.2 Lack of considering building geometry interaction

Building geometries are usually complex for large-span buildings. The complexity is reflected not only in a single type of building components, but also in the interaction of multiple types of building components. For instance, interior functional space can interact with overall building envelope geometries; overall roof geometries can interact with roof structure geometries and roof cladding geometries (Turrin et al., 2012). When it comes to sports buildings in particular, various kinds of geometries need to be dealt with, including grandstands, building envelopes (e.g., glazing, walls and roofs) and roof structure geometries. These geometries often interact with each other. For example, a small adjustment to a grandstand (e.g., slightly raising the first row or changing its curvature, creating a bit more space between rows) can have huge impacts on the overall size and geometry of the building envelope and roof structure. Such interaction can be found during the conceptual design of outdoor sports buildings, such as the Allianz Arena (2006), the Beijing National Stadium (2009), and Singapore Sports Hub (2015); and it is also true for indoor sports buildings.

Despite the complexity of building geometries, current conceptual sports building design studies that utilize optimal-design methods have not fully considered building geometry interaction. They mainly focus on the geometry of a single type of building components, such as grandstand geometry (Zargar and Alaghmandan, 2019; Bianconi et al., 2020), building envelope geometry (Shi and Yang, 2013), and roof structure geometry (Arkinstall and Carfrae, 2006; Holzer et al., 2007; Flager et al., 2009b). In contrast, only a few studies focus on the integration of multiple types of building geometries, such as the integration of building envelope and roof structure geometries (Brown and Mueller, 2016), and the integration of grandstand, building envelope and roof structure geometries (Yang et al., 2018; Pan et al., 2019). Thus, it is worth highlighting building geometry interaction in conceptual sports building design studies that utilize optimal-design methods.

2.6.2.3 Lack of consideration building performance conflicts

Building performances are usually complex for large-span buildings. The complexity is reflected not only in a single type of building performance but also in the conflicts of multiple types of building performances. For instance, the maximization of daylight availability can conflict with the minimization of operational energy use (Lartigue et al., 2014; Manzan and Clarich, 2017; Futrell et al., 2015); the minimization of structural weight can conflict with the minimization of operational energy use (Flager et al., 2009b); the minimization of structural displacement can conflict with the minimization of structural weight (Papadrakakis et al., 2002); and the architectural aesthetics can conflict with other engineering performances (Mueller and Ochsendorf, 2015). When it comes to sports buildings in particular, various kinds of performances need to be considered, including architectural, climatic, structural, and energy performances. These performances are often in conflict with each other. For example, a small adjustment to a grandstand (e.g., slightly raising the first row or changing its curvature, creating a bit more space between rows) may provide better view quality for spectators, but lead to a larger structure that demands more embodied energy and construction costs. Such conflicts can be found during the conceptual design of outdoor sports buildings, such as the Allianz Arena (2006), the Beijing National Stadium (2009), and Singapore Sports Hub (2015); and it is also true for indoor sports buildings.

Despite the complexity of building performances, current conceptual sports building design studies that utilize optimal-design methods have not fully considered building performance conflicts. They mainly focus on a single type of building performance, such as view quality performance (Zargar and Alaghmandan, 2019; Bianconi et al., 2020), solar radiation performance (Shi and Yang, 2013); and structural performance (Arkinstall and Carfrae, 2006; Holzer et al., 2007; Flager et al., 2009b). In contrast, only a few studies focus on the conflicts of multiple types of building performances, such as the conflict of embodied and operational energy performances (Brown and Mueller, 2016), the conflict of daylighting, operational energy, and structural performances (Yang et al., 2018), and the conflict of view quality, structural and acoustic performances (Pan et al., 2019). Thus, it is worth highlighting building performance conflicts in conceptual sports building design studies that utilize optimal-design methods.

2.7 Conclusion

This chapter concludes by summarizing the main research results (Section 2.7.1) and providing concluding remarks (Section 2.7.2).

2.7.1 Main research results

The main research results of this chapter include the following:

- Dynamic and interactive re-definition or re-formulation has been identified as a potential means to achieve a more reliable design task and optimization problem. It highlights continuous knowledge extraction, quantitative and qualitative thinking, and divergent and convergent thinking. However, it has not often been discussed in optimal-design methods.

- Dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) has rarely been incorporated in Multi-Objective Optimization (MOO) design methods. The few methods that can do so still have room for improvement, especially in terms of information and knowledge extraction.

- Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software have rarely been integrated into Multi-Objective Optimization (MOO) software workflows. A software workflow that can do so still has limitations, especially in terms of software integration.

- Discussions about optimal-design methods for conceptual sports building design have increased recently. Nevertheless, related studies have not fully considered the complexity of building geometry and building performance; and Multi-Objective Optimization (MOO) design methods have not been commonly applied to this design field.

2.7.2 Concluding remarks

In conclusion, it is necessary to establish: a new Multi-Objective and Multi-Disciplinary Optimization (MOMDO) design method that can incorporate dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) (to be elaborated

in Chapter 3); and an improved Multi-Objective and Multi-Disciplinary Optimization (MOMDO) software workflow where Grasshopper and modeFRONTIER are integrated in a better manner (to be elaborated in Chapter 4). Moreover, it is valuable to apply the proposed design method and software workflow to the conceptual design of indoor sports halls (to be elaborated in Chapters 5 and 6).

References

Agte, J., De Weck, O., Sobieszczanski-Sobieski, J., Arendsen, P., Morris, A., & Spieck, M. (2010). MDO: assessment and direction for advancement - an opinion of one international group. Structural and Multidisciplinary Optimization, 40(1-6), 17. DOI: https://doi.org/10.1007/s00158-009-0381-5.

Aish, R., & Woodbury, R. (2005). Multi-level interaction in parametric design. In: A. Butz, B. Fisher, A. Krüger, & P. Olivier (Eds.), Smart Graphics (pp. 151-162). Springer, Berlin, Heidelberg. ISBN: 978-3-540-28179-5.

American Institute of Architects (2017). B101- 2017 Standard Form of Agreement Between Owner and Architect. https://www.aiacontracts.org/contract-documents/25141-owner-architect-agreement [Accessed: 24-September-2021].

Andersson, J. (2000). A survey of multiobjective optimization in engineering design. Technical Report, Linktjping University, Sweden. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=7DDF-6CE2B9F8F2C42D13381870BF151E?doi=10.1.1.8.5638&rep=rep1&type=pdf [Accessed: 24-September-2021].

Arkinstall, M., & Carfrae, T. (2006). Structural Design and Optimisation of the Beijing National Olympic Swimming Centre. Australian Journal of Structural Engineering, 6(3), 181-190. DOI: https://doi.org/10.1080/13287982.2006.11464954.

Arora, J. S. (1990). Computational design optimization: a review and future directions. Structural Safety, 7(2-4), 131-148. DOI: https://doi.org/10.1016/0167-4730(90)90063-U.

Arora, J. S. (2016). Introduction to Optimum Design. Fourth Edition. Academic Press, Cambridge. ISBN: 9780128008065.

Azzouz, R., Bechikh, S., & Said, L. B. (2017). Dynamic multi-objective optimization using evolutionary algorithms: a survey. In: B. Slim, D. Rituparna, & G. Abhishek (Eds.), Recent Advances in Evolutionary Multi-objective Optimization (pp. 31-70). Springer, Cham. ISBN: 978-3-319-42978-6.

Ball, L. J., Evans, J. S. B., & Dennis, I. (1994). Cognitive processes in engineering design: a longitudinal study. Ergonomics, 37(11), 1753-1786. DOI: https://doi.org/10.1080/00140139408964950.

Barnum, G. J., & Mattson, C. A. (2010). A computationally assisted methodology for preference-guided conceptual design. Journal of Mechanical Design, 132 (12), 121003. DOI: https://doi.org/10.1115/1.4002838.

Becker, R. (2008). Fundamentals of performance-based building design. Building Simulation, 1(4), 356-371. DOI: https://doi.org/10.1007/s12273-008-8527-8.

Bernal, M., Haymaker, J. R., & Eastman, C. (2015). On the role of computational support for designers in action. Design Studies, 41, 163-182. DOI: https://doi.org/10.1016/j.destud.2015.08.001.

Bianconi, F., Filippucci, M., Buffi, A., & Vitali, L. (2020). Morphological and visual optimization in stadium design: a digital reinterpretation of Luigi Moretti's stadiums. Architectural Science Review, 63(2), 194-209. DOI: https://doi.org/10.1080/00038628.2019.1686341.

Blanchard, B. S., & Fabrycky, W. J. (2011). Systems Engineering and Analysis. Fifth Edition. Prentice Hall, Hoboken. ISBN: 978-0132217354.

Boshernitsan, M., & Downes, M. S. (2004). Visual programming languages: a survey. Report of Computer Science Division, University of California, USA. http://citeseerx.ist.psu.edu/viewdoc/download?-doi=10.1.1.73.2491&rep=rep1&type=pdf [Accessed: 24-September-2021].

Brintrup, A. M., Ramsden, J., & Tiwari, A. (2007). An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. Computers in Industry, 58(3), 279-291. DOI: https://doi.org/10.1016/j.compind.2006.06.004.

Brown, N., De Oliveira, J. I. F., Ochsendorf, J., & Mueller, C. (2016). 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 (pp. 1103-1111). CRC Press, Leiden. ISBN: 978-1138026513.

Brown, N. C., & Mueller, C. T. (2016). Design for structural and energy performance of long span buildings using geometric multi-objective optimization. Energy and Buildings, 127, 748-761. DOI: https://doi.org/10.1016/j.enbuild.2016.05.090.

Building Code Committee (1925). Recommended practice for arrangement of building codes. Report of Building Code Committee. U.S. Government Printing Office, Washington.

Caldas, L. (2001). An Evolution-based Generative Design System: Using Adaptation to Shape Architectural Form (doctoral dissertation). Department of Architecture, Massachusetts Institute of Technology, USA.

Caldas, L. (2006). GENE_ARCH: an evolution-based generative design system for sustainable architecture. In: I. F. C. Smith (Ed.), Intelligent Computing in Engineering and Architecture (pp. 109-118). Springer, Berlin, Heidelberg. ISBN: 978-3-540-46246-0.

Caldas, L. (2008). Generation of energy-efficient architecture solutions applying GENE_ARCH: An evolution-based generative design system. Advanced Engineering Informatics, 22(1), 59-70. DOI: https://doi.org/10.1016/j.aei.2007.08.012.

Caldas, L. G., & Norford, L. K. (2002). A design optimization tool based on a genetic algorithm. Automation in construction, 11(2), 173-184. DOI: https://doi.org/10.1016/S0926-5805(00)00096-0.

Caldas, L. G., & Norford, L. K. (2003). Shape generation using pareto genetic algorithms: integrating conflicting design objectives in low-energy architecture. International Journal of Architectural Computing, 1(4), 503-515. DOI: https://doi.org/10.1260/147807703773633509.

Chantrelle, F. P., Lahmidi, H., Keilholz, W., El Mankibi, M., & Michel, P. (2011). Development of a multicriteria tool for optimizing the renovation of buildings. Applied Energy, 88(4), 1386-1394. DOI: https://doi.org/10.1016/j.apenergy.2010.10.002.

Cheng, Z., Li, L., & Bahnfleth, W. P. (2016). Natural ventilation potential for gymnasia–Case study of ventilation and comfort in a multisport facility in northeastern United States. Building and Environment, 108, 85-98. DOI: https://doi.org/10.1016/j.buildenv.2016.08.019.

Cichocka, J. M., Browne, W. N., & Rodriguez, E. (2017). Optimization in the architectural practice. 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 (pp. 387-396). CAADRIA and Xi'an Jiaotong-Liverpool University. ISBN: 978-988-19026-8-9.

Conti Z X. (2013). Multi-Objective Optimisation of Building Geometry for Energy Consumption and View Quality (master thesis). Department of Architecture and Civil Engineering, University of Bath, UK.

Conti, Z. X., Shepherd, P., & Richens, P. (2015). 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 (pp. 287-294). eCAADe and Vienna University of Technology. ISBN: 9789491207082.

Cross, N. (2008). Engineering Design Methods: Strategies for Product Design. Fourth Edition. John Wiley & Sons, Hoboken. ISBN: 978-0-470-51926-4.

Culley, P., & Pascoe, J. (2009). Sports Facilities and Technologies. Routledge, New York. ISBN: 978-0-415-45868-9.

CURT. (2004). Collaboration, integrated information and the project lifecycle in building design, construction and operation. White Paper of the Construction Users Roundtable. https://kcuc.org/wp-content/uploads/2013/11/Collaboration-Integrated-Information-and-the-Project-Lifecycle.pdf [Accessed: 24-September-2021].

Curtis, S. K., Mattson, C. A., Hancock, B. J., & Lewis, P. K. (2013). Divergent exploration in design with a dynamic multiobjective optimization formulation. Structural and Multidisciplinary Optimization, 47(5), 645-657. DOI: https://doi.org/10.1007/s00158-012-0855-8.

Danhaive, R. A., & Mueller, C. T. (2015). Combining parametric modeling and interactive optimization for high-performance and creative structural design. In: Proceedings of IASS Annual Symposia, IASS 2015 Amsterdam Symposium: Future Visions – Computational Design (pp. 1-11). IASS. ISSN: 2518-6582.

Deb, K. (2014). Multi-objective optimization. In: E. K. Burke, & G. Kendall (Eds.), Search methodologies (pp. 403-449). Springer, Boston. ISBN: 978-1-4614-6939-1.

Ding, F. (2017). Daylight integration and visual comfort in sports halls in Norway (master thesis). Faculty of Architecture and Design, Norwegian University of Science and Technology, Norway.

Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem–solution. Design studies, 22(5), 425-437. DOI: https://doi.org/10.1016/S0142-694X(01)00009-6.

Ekici, B., Cubukcuoglu, C., Turrin, M., & Sariyildiz, I. S. (2019). Performative computational architecture using swarm and evolutionary optimisation: A review. Building and Environment, 147, 356-371. DOI: https://doi.org/10.1016/j.buildenv.2018.10.023.

Eschenauer, H. A., Schuhmacher, G., & Hartzheim, W. (1992). Multidisciplinary design of composite aircraft structures by lagrange. Computers & Structures, 44(4), 877-893. DOI: https://doi.org/10.1016/0045-7949(92)90475-F.

Evins, R. (2013). A review of computational optimisation methods applied to sustainable building design. Renewable and Sustainable Energy Reviews, 22, 230-245. DOI: https://doi.org/10.1016/j.rser.2013.02.004.

Fabrycky, W. J. (1994). Modeling and indirect experimentation in system design evaluation. Systems Engineering, 1(1), 133-144. https://a2i2.com/download/modeling-and-indirect-experimentation-in-system-design-evaluation-incose-1994/ [Accessed: 24-September-2021].

Fabrycky, W. J., & Blanchard, B. S. (1991). Life Cycle Cost and Economic Analysis. Prentice Hall, Hoboken. ISBN 978-0135383230.

Flager, F., Adya, A., & Haymaker, J. (2009b). AEC multidisciplinary design optimization: Impact of high performance computing. Technical Report, Stanford University, USA. https://purl.stanford.edu/rb200qx5476 [Accessed: 24-September-2021].

Flager, F., Welle, B., Bansal, P., Soremekun, G., & Haymaker, J. (2009a). Multidisciplinary process integration and design optimization of a classroom building. Journal of Information Technology in Construction (ITcon), 14(38), 595-612. http://www.itcon.org/2009/38 [Accessed: 24-September-2021].

Fontan, J. (2021). 5 Architecture Phases of Design Explained. https://fontanarchitecture.com/architectural-design-phases/ [Accessed: 24-September-2021].

Futrell, B. J., Ozelkan, E. C., & Brentrup, D. (2015). Bi-objective optimization of building enclosure design for thermal and lighting performance. Building and Environment, 92, 591-602. DOI: https://doi.org/10.1016/j.buildenv.2015.03.039.

Gaetani, I., Hoes, P. J., & Hensen, J. L. (2020). A stepwise approach for assessing the appropriate occupant behaviour modelling in building performance simulation. Journal of Building Performance Simulation, 13(3), 362-377. DOI: https://doi.org/10.1080/19401493.2020.1734660.

Gagne, J. M., & Andersen, M. (2010). Multi-objective facade optimization for daylighting design using a genetic algorithm. In: Proceedings of SimBuild 2010: the Fourth National Conference on the International Building Performance Simulation Association (pp. 110-117). IBPSA-USA. https://infoscience.epfl.ch/record/153674 [Accessed: 24-September-2021].

Gerber, D., & Lin, S. H. E. (2012). Designing-in performance through parameterisation, automation, and evolutionary algorithms: 'HDS BEAGLE 1.0'[A]. In: T. Fischer, K. DeBiswas, 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 (pp. 141-150). CAADRIA and Hindustan University. ISBN: 978-988-19026-3-4.

Gerber, D. J., & Lin, S. H. E. (2014). Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture. Simulation, 90(8), 936-959. DOI: https://doi.org/10.1177/0037549713482027.

Gero, J. S. (1994). Towards a model of exploration in computer-aided design. In: J. S. Gero, & E. Tyugu (Eds.), Formal Design Methods for CAD: Proceedings of the IFIP TC5/WG5.2 Workshop on Formal Design Methods for CAD (pp. 315–336). Elsevier Science, Amsterdam. ISBN: 978-0-444-81970-3.

Gibson, E. J. (1982). Working with the performance approach in building. CIB Report Publication 64. CIB, Rotterdam.

Graf, J. (1995). Interactive Evolutionary Algorithms in Design. In: D.W. Pearson, N.C. Steele, & R.F. Albrecht (Eds.), Artificial Neural Nets and Genetic Algorithms (pp. 227-230). Springer, Vienna. ISBN: 978-3-211-82692-8

Graf, J. (1996). Interactive evolution: A prototyping framework for engineering. Computer-Aided Civil and Infrastructure Engineering, 11(3), 185-193. DOI: https://doi.org/10.1111/j.1467-8667.1996.tb00322.x.

Gross, J. G. (1996). Keynote Speech: Developments in the application of the performance concept in building. In: R. Becker, & M. Paciuk (Eds.), Proc. 3rd CIB-ASTM-ISO-RILEM International Symposium (Vol. 1) (pp. 1-11). National Building Research Institute, Haifa. DOI: https://doi.org/10.1007/BF02486358.

Hamdy, M., Hasan, A., & Siren, K. (2011). Impact of adaptive thermal comfort criteria on building energy use and cooling equipment size using a multi-objective optimization scheme. Energy and Buildings, 43(9), 2055-2067. DOI: https://doi.org/10.1016/j.enbuild.2011.04.006.

Harpster, M. (2021). The Architectural Design Process Explained. https://layer.team/the-architectural-design-process-explained/ [Accessed: 24-September-2021].

Hattis, D. B., & Becker, R. (2001). Comparison of the systems approach and the Nordic model and their melded application in the development of performance-based building codes and standards. Journal of Testing and Evaluation, 29(4), 413-422. DOI: https://doi.org/10.1520/JTE12271J.

Hay, L., Duffy, A. H., McTeague, C., Pidgeon, L. M., Vuletic, T., & Grealy, M. (2017). A systematic review of protocol studies on conceptual design cognition: Design as search and exploration. Design Science, 3(e10). DOI: https://doi.org/10.1017/dsj.2017.11.

Heinzelmann, F. (2018). Design method for adaptive daylight systems for buildings covered by large (span) roofs (doctoral dissertation). Department of the Built Environment, Eindhoven University of Technology, the Netherlands.

Heiselberg, P., Brohus, H., Hesselholt, A., Rasmussen, H., Seinre, E., & Thomas, S. (2009). Application of sensitivity analysis in design of sustainable buildings. Renewable Energy, 34(9), 2030-2036. DOI: https://doi.org/10.1016/j.renene.2009.02.016.

Helbig, M., & Engelbrecht, A. P. (2013). Dynamic multi-objective optimization using PSO. In: E. Alba, A. Nakib, & P. Siarry (Eds.), Metaheuristics for Dynamic Optimization (pp. 147-188). Springer, Berlin, Heidelberg. ISBN: 978-3-642-30664-8.

Helbig, M., & Engelbrecht, A. P. (2014). Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. Swarm and Evolutionary Computation, 14, 31-47. DOI: https://doi.org/10.1016/j.swevo.2013.08.004.

Hensen, J. L. M., & Lamberts, R. (Eds.). (2011). Building Performance Simulation for Design and Operation. Spon Press, Abingdon, New York. ISBN: 978-0-415-47414-6.

Hien, W. N., Poh, L. K., & Feriadi, H. (2000). The use of performance-based simulation tools for building design and evaluation - a Singapore perspective. Building and Environment, 35(8), 709-736. DOI: https://doi.org/10.1016/S0360-1323(99)00059-1.

Holzer, D., Hough, R., & Burry, M. (2007). Parametric design and structural optimisation for early design exploration. International Journal of Architectural Computing, 5(4), 625-643. DOI: https://doi.org/10.1260/147807707783600780.

Horváth, I., (2004) On some crucial issues of computer support of conceptual design. In: D. Talaba, & T. Roche (Eds.), Product engineering: eco-design, technologies and green energy (pp. 123-142). Springer, Dordrecht. ISBN: 978-1-4020-2932-5.

Huang, Y., & Niu, J. L. (2016). Optimal building envelope design based on simulated performance: History, current status and new potentials. Energy and Buildings, 117, 387-398. DOI: https://doi.org/10.1016/j.enbuild.2015.09.025.

Huovila, P., Boswell, P., Çölaşan, F., Félix, J., Jenkins, S., Nadar, A., Van der Putte, I., Read, A., Záhonyi, Z., Halcomb, J., & Yaker, F. (2018). Guidance document on procuring sustainable buildings and construction. Report of UN Environment Programme. UN Environment, Paris.

Janssen P. (2013). 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, Volume 1 (pp. 205-214). eCAADe and Delft University of Technology. ISBN: 978-94-91207-04-4.

Janssen, P. (2015). Dexen: A scalable and extensible platform for experimenting with population-based design exploration algorithms. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 29(4), 443-455. DOI: https://doi.org/10.1017/S0890060415000438.

Janssen, P., Basol, C., & Chen, K. W. (2011). Evolutionary developmental design for non-programmers. In: T. 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 (pp. 245-252). eCAADe and University of Ljubljana. ISBN: 978-9-4912070-1-3.

John, G., & Heard, H. (1981). Handbook of Sports and Recreational Building Design, Volume 2 Indoor Sports. The Architectural Press, London. ISBN: 978-0851395876.

Jonas, W. (1993). Design as problem-solving? or: Here is the solution - What was the problem? Design Studies, 14(2), 157-170. DOI: https://doi.org/10.1016/0142-694X(93)80045-E

Jones, J. C. (1992). Design methods. Second Edition. John Wiley & Sons, Hoboken. ISBN: 978-0471284963.

Josa, I., Pons, O., de la Fuente, A., & Aguado, A. (2020). Multi-criteria decision-making model to assess the sustainability of girders and trusses: Case study for roofs of sports halls. Journal of Cleaner Production, 249, 119312. DOI: https://doi.org/10.1016/j.jclepro.2019.119312.

Joseph, D., Kim, A., Butler, A., & Haeusler, M. H. (2015). Optimisation for Sport Stadium Designs. In: Y. Ikeda, C. M. Herr, D. Holzer, S. Kaijima, M. J. Kim, & M. A. Schnabel (Eds.), Emerging Experience in Past, Present and Future of Digital Architecture, Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (pp. 573-582). CAADRIA and Kyungpook National University. ISBN: 978-988-19026-6-5.

Kanaani, M. (2015). Performativity: the fourth dimension in architectural design. In: M. Kanaani, & D. Kopec (Eds.), The Routledge Companion for Architecture Design and Practice (pp. 125-148). Routledge, Oxfordshire. ISBN: 978-1-138-02315-4.

Kaushik, V., & Janssen, P. (2013). 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 (pp. 137-146). CAADRIA and National University of Singapore. ISBN: 9789881902641.

King, L. W. (2004). The Code of Hammurabi. Kessinger Publishing, Whitefish. ISBN: 978-1419157035.

Kolarevic, B., (2003a), 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 (pp. 457-464). eCAADe and Graz University of Technology. ISBN: 0-9541183-1-6.

Kolarevic, B. (2003b). Digital morphogenesis. In: B. Kolarevic (Ed.), Architecture in the Digital Age: Design and Manufacturing (pp. 17-45). Spon Press, New York. ISBN: 0-415-27820-1.

Kolarevic, B. (2004). Back to the future: performative architecture. International Journal of Architectural Computing, 2(1), 43-50. DOI: https://doi.org/10.1260/1478077041220205.

Kolarevic, B. (2005). Computing the performative. In: B. Kolarevic, & A. M. Malkawi (Eds.), Performative Architecture: Beyond Instrumentality (pp. 193-202). Spon Press, New York. ISBN: 978-0415700832.

Lartigue, B., Lasternas, B., & Loftness, V. (2014). Multi-objective optimization of building envelope for energy consumption and daylight. Indoor and Built Environment, 23(1), 70-80. DOI: http://dx.doi.org/10.1177/1420326X13480224.

Leach, N. (2009). Digital morphogenesis. Architectural Design, 79(1), 32-37. DOI: https://doi.org/10.1002/ad.806.

Liu, Y. C., Chakrabarti, A., & Bligh, T. (2003). Towards an 'ideal' approach for concept generation. Design studies, 24(4), 341-355. DOI: https://doi.org/10.1016/S0142-694X(03)00003-6.

Logan, B., & Smithers, T. (1993). Creativity and design as exploration. In: J. S. Gero, & M. L. Maher (Eds.), Modeling Creativity and Knowledge-based Creative Design (pp. 139-173). Lawrence Erlbaum Associates, Hillsdale. ISBN: 0-8058-1153-2.

Maher, M. L., & Poon, J. (1996). Modeling design exploration as co-evolution. Computer-Aided Civil and Infrastructure Engineering, 11(3), 195-209. DOI: https://doi.org/10.1111/j.1467-8667.1996.tb00323.x

Maher, M. L., Poon, J., & Boulanger, S. (1996). Formalising design exploration as co-evolution. In: J. S. Gero, & F. Sudweeks (Eds.), Advances in formal design methods for CAD: Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for CAD (pp. 3-30). Springer Science + Business Media, Dordrecht. ISBN: 978-1-4757-4428-6.

Malkawi, A. M. (2005). Performance simulation: research and tools. In: B. Kolarevic, & A. M. Malkawi (Eds.), Performative Architecture: Beyond Instrumentality (pp. 85-96). Spon Press, New York. ISBN: 978-0415700832.

Manzan, M., & Clarich, A. (2017). FAST energy and daylight optimization of an office with fixed and movable shading devices. Building and Environment, 113, 175-184. DOI: http://dx.doi.org/10.1016/j.buildenv.2016.09.035.

Marconi, C. (Ed.). (2015). The Oxford handbook of Greek and Roman art and architecture. Oxford University Press, New York. ISBN: 9780199783304.

Mavris, D. N., & DeLaurentis, D. A. (2000). Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design. In: I. Grant (Ed.), CD-ROM proceedings of 22nd International Congress of Aeronautical Sciences (1.4.4). Optimage, Edinburgh. ISBN 0-9533991-2-5.

Mavris, D. N., DeLaurentis, D. A., Bandte, O., & Hale, M. (1998). A stochastic approach to multi-disciplinary aircraft analysis and design. In: 36th AIAA Aerospace Sciences Meeting and Exhibit (pp. 1-15). AIAA, Reston. DOI: https://doi.org/10.2514/6.1998-912.

Meignan, D., Knust, S., Frayret, J. M., Pesant, G., & Gaud, N. (2015). A review and taxonomy of interactive optimization methods in operations research. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(3), 1-43. DOI: https://doi.org/10.1145/2808234.

Meniru, K., Rivard, H., & Bédard, C. (2003). Specifications for computer-aided conceptual building design. Design Studies, 24(1), 51-71. DOI: https://doi.org/10.1016/S0142-694X(02)00009-1.

Mueller, C. T., & Ochsendorf, J. A. (2015). Combining structural performance and designer preferences in evolutionary design space exploration. Automation in Construction, 52, 70-82. DOI: https://doi.org/10.1016/j.autcon.2015.02.011.

Navinchandra, D. (2011). Exploration and Innovation in Design: Towards a Computational Model. Springer-Verlag, New York. ISBN: 978-1-4612-7802-3.

Negendahl, K., & Nielsen, T. R. (2015). Building energy optimization in the early design stages: A simplified method. Energy and Buildings, 105, 88-99. DOI: https://doi.org/10.1016/j.enbuild.2015.06.087.

Newton, D. (2018). Accommodating change and open-ended search in design optimization. In: T. Fukuda, W. Huang, P. Janssen, K. Crolla, & S. Alhadidi (Eds.), Learning, Adapting and Prototyping: Proceedings of the 23rd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (pp. 175-184). CAADRIA and Tsinghua University. ISBN: 978-988-19026-0-3.

Nguyen, A. T., Reiter, S., & Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. Applied Energy, 113, 1043-1058. DOI: https://doi.org/10.1016/j.apenergy.2013.08.061.

Nord, N., Mathisen, H. M., & Cao, G. (2015). Energy cost models for air supported sports hall in cold climates considering energy efficiency. Renewable Energy, 84, 56-64. DOI: https://doi.org/10.1016/j.renene.2015.07.003.

Oliver, P. (2006). Built to meet needs: Cultural issues in vernacular architecture. Routledge. Architectural Press, New York. ISBN: 978-0750666572.

Oxman, R. (2008a). Emerging paradigms and models in digital design - performance-based architectural design. In: P. Brandon, & T. Kocatürk (Eds.), Virtual Futures for Design, Construction and Procurement (pp. 3-20). Blackwell Publishing, Oxford. ISBN: 978-1-4051-7024-6.

Oxman, R. (2008b). Performance-based design: current practices and research issues. International Journal of Architectural Computing, 6(1), 1-17. DOI: https://doi.org/10.1260/147807708784640090.

Oxman, R., Hammer, R., & Ari, S. B. (2007). 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 (pp. 227-234). eCAADe and FH Frankfurt. ISBN: 978-0-9541183-6-5.

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. (2007). Engineering Design: A Systematic Approach. Third Edition. Springer-Verlag, London. ISBN: 978-1-84628-318-5.

Palonen, M., Hamdy, M., & Hasan, A. (2013). MOBO a new software for multi-objective building performance optimization. In: Proceedings of Building Simulation 2013 - 13th Conference of International Building Performance Simulation Association (pp. 2567-2574). IBPSA. ISSN: 2522-2708.

Pan, W., Turrin, M., Louter, C., Sariyildiz, S., & Sun, Y. (2019). Integrating multi-functional space and long-span structure in the early design stage of indoor sports arenas by using parametric modelling and multi-objective optimization. Journal of Building Engineering, 22, 464-485. DOI: https://doi.org/10.1016/j.jobe.2019.01.006.

Papadrakakis, M., Lagaros, N., & Plevris, V. (2002). Multi-objective optimization of skeletal structures under static and seismic loading conditions. Engineering Optimization, 34(6), 645-669. DOI: https://doi.org/10.1080/03052150215716.

Paulson Jr, B. C. (1976). Designing to reduce construction costs. Journal of the Construction Division, 102(4), 587-592. DOI: https://doi.org/10.1061/JCCEAZ.0000639.

Porritt, S. M., Cropper, P. C., Shao, L., & Goodier, C. I. (2012). Ranking of interventions to reduce dwelling overheating during heat waves. Energy and Buildings, 55, 16-27. DOI: https://doi.org/10.1016/j.enbuild.2012.01.043.

Pugh, S. (1991). Total Design: Integrated Methods for Successful Product Engineering. Addison-Wesley, Boston. ISBN: 978-0201416398.

Rajagopalan, P., & Luther, M. B. (2013). Thermal and ventilation performance of a naturally ventilated sports hall within an aquatic centre. Energy and buildings, 58, 111-122. DOI: https://doi.org/10.1016/j.enbuild.2012.11.022.

Raquel, C., & Yao, X. (2013). Dynamic multi-objective optimization: a survey of the state-of-the-art. In: S. Yang, & X. Yao (Eds.), Evolutionary Computation for Dynamic Optimization Problems (pp. 85-106). Springer, Berlin, Heidelberg. ISBN: 978-3-642-38415-8.

Reisner-Cook, Y. (2009). The troubled relationship between architecture and aesthetic: exploring the self and emotional beauty in design (doctoral dissertation). School of Architecture and Design, Royal Melbourne Institute of Technology, Australia.

Riccobono, A., Koutamanis, A., & Pellitteri, G. (2013). Digital Expressionism: the architecture of complex shapes. In: M. A. Schnabel (Ed.), Cutting Edge: 47th International Conference of the Architectural Science Association (pp. 663-672). The Architectural Science Association, Australia. ISBN: 978-0-9923835-0-3.

Rowe, P. G. (1991). Design thinking. MIT press, Cambridge. ISBN: 978-0262680677.

Sarangam, A. (2020). Data Analysis vs Analytics: All You Need to Know in 2020. https://www.jigsawacademy.com/blogs/business-analytics/analysis-vs-analytics/ [Accessed: 24-September-2021].

Sariyildiz, S. (2012). Keynote Speech: Performative Computational Design. In: ICONARCH-I "Architecture & Technology" International Congress: Proceedings Book (pp. 313-344). Selçuklu Municipality of Konya, Konya. ISBN: 978-975-448-203-4.

Schneider, L. (2018). 6 Phases of the Design Process. https://www.schooleycaldwell.com/blog/architecture-explained-the-6-phases-of-design [Accessed: 24-September-2021].

Shea, K., Sedgwick, A., & Antonuntto, G. (2006). Multicriteria optimization of paneled building envelopes using ant colony optimization. In: I. F. C. Smith (Ed.), Intelligent Computing in Engineering and Architecture (pp. 627-636). Springer, Berlin, Heidelberg. ISBN: 978-3-540-46246-0.

Shen, H., & Tzempelikos, A. (2013). Sensitivity analysis on daylighting and energy performance of perimeter offices with automated shading. Building and Environment, 59, 303-314. DOI: https://doi.org/10.1016/j.buildenv.2012.08.028.

Shi, X., & Yang, W. (2013). Performance-driven architectural design and optimization technique from a perspective of architects. Automation in Construction, 32, 125-135. DOI: https://doi.org/10.1016/j.autcon.2013.01.015.

Simpson, T. W., & Martins, J. R. R.A. (2011). Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop. Journal of Mechanical Design, 133 (10), 101002. DOI: https://doi.org/10.1115/1.4004465.

Smith, R. P., & Tjandra, P. (1998). Experimental observation of iteration in engineering design. Research in Engineering Design, 10(2), 107-117. DOI: https://doi.org/10.1007/BF01616691.

Smithers, T. (1992). Design as exploration: puzzle-making and puzzle-solving. https://www.academia.edu/22991882/Design_as_Exploration_Puzzle-Making_and_Puzzle-Solving [Accessed: 24-September-2021].

Smithers, T., Corne, D., & Ross, P. (1994). On computing exploration and solving design problems. In: J. S. Gero, & E. Tyugu (Eds.), Formal Design Methods for CAD: Proceedings of the IFIP TC5/WG5.2 Workshop on Formal Design Methods for CAD (pp. 293-313). Elsevier Science, Amsterdam. ISBN: 978-0-444-81970-3.

Smithers, T., & Troxell, W. (1990). Design is intelligent behaviour, but what's the formalism? Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 4(2), 89-98. DOI: https://doi.org/10.1017/S0890060400002286.

Sun, Y., Xiong, L., & Su, P. (2013). Grandstand Grammar and its Computer Implementation. In: R. Stouffs, & S. Sariyildiz (Eds.), Computation and Performance: Proceedings of the 31st Conference on Education in Computer Aided Architectural Design in Europe, Volume 2 (pp. 645-654). eCAADe and Delft University of Technology. ISBN: 978-94-91207-05-1.

Suo, H., Angelotti, A., & Zanelli, A. (2015). Thermal-physical behavior and energy performance of air-supported membranes for sports halls: A comparison among traditional and advanced building envelopes. Energy and Buildings, 109, 35-46. DOI: https://doi.org/10.1016/j.enbuild.2015.10.011.

Szigeti, F., & Davis, G. (2005). Performance based building: conceptual framework. CIB PeBBu Final Report. CIB, Rotterdam. ISBN: 90-6363-051-4.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. Proceedings of the IEEE, 89(9), 1275-1296. DOI: https://doi.org/10.1109/5.949485.

Thornton Tomasetti (2017). Sports + Public Assembly. Report of Thornton Tomasetti. https://www.thornton-tomasetti.com/resource/sports-public-assembly [Accessed: 24-September-2021].

Touloupaki, E., & Theodosiou, T. (2017). Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization—A review. Energies, 10(5), 637. DOI: https://doi.org/10.3390/en10050637.

Trabelsi, W., Azzouz, R., Bechikh, S., & Said, L. B. (2016). Leveraging evolutionary algorithms for dynamic multi-objective optimization scheduling of multi-tenant smart home appliances. In: 2016 IEEE Congress on Evolutionary Computation (CEC) (pp. 3533-3540). IEEE, Piscataway. ISBN: 978-1-5090-0622-9.

Turrin, M., Von Buelow, P., Kilian, A., & Stouffs, R. (2012). Performative skins for passive climatic comfort: A parametric design process. Automation in Construction, 22, 36-50. DOI: https://doi.org/10.1016/j.autcon.2011.08.001.

Turrin, M., Von Buelow, P., & Stouffs, R. (2011). Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. Advanced Engineering Informatics, 25(4), 656-675. DOI: https://doi.org/10.1016/j.aei.2011.07.009.

Ullman, D. G., Dietterich, T. G., & Stauffer, L. A. (1988). A model of the mechanical design process based on empirical data. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2(1), 33-52. DOI: https://doi.org/10.1017/S0890060400000536.

Ulrich, K. T., & Eppinger, S. D. (2012). Product design and development. Fifth Edition. McGraw-Hill, New York. ISBN: 9780073404776.

Vierlinger, R., & Bollinger, K. (2014). Acommodating Change in Parametric Design. In: D. J. Gerber, A. Huang, & J. Sanchez (Eds.), Design Agency: Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture (pp. 609-618). Riverside Architectural Press, Cambridge. ISBN: 9781926724478

Von Buelow, P. (2012). ParaGen: Performative Exploration of generative systems. Journal of the International Association for Shell and Spatial Structures, 53(4), 271-284. ISSN: 1028-365X.

Von Buelow, P. (2016). Genetically enhanced parametric design in the exploration of architectural solutions. In: P. J. S. Cruz (Ed.), Structures and Architecture - Beyond Their Limits (pp. 675-683). CRC Press, Leiden. ISBN: 978-1138026513.

Wright, J. A., Brownlee, A., Mourshed, M. M., & Wang, M. (2014). Multi-objective optimization of cellular fenestration by an evolutionary algorithm. Journal of Building Performance Simulation, 7(1), 33-51. DOI: https://doi.org/10.1080/19401493.2012.762808.

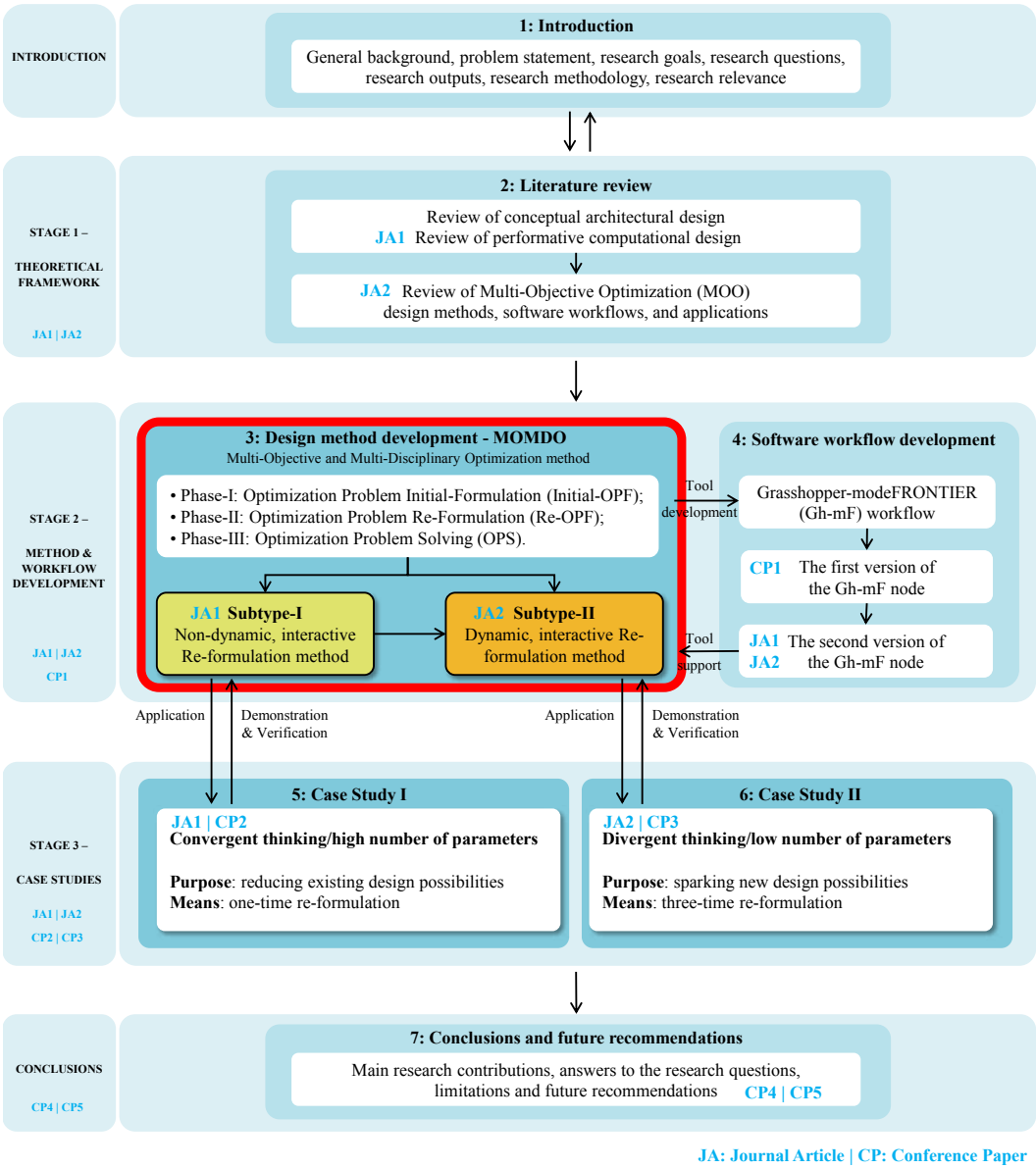
Yang, D., Di Stefano, D., Turrin, M., Sariyildiz, S., & Sun, Y. (2020). Dynamic and interactive re-formulation of multi-objective optimization problems for conceptual architectural design exploration. Automation in Construction, 118, 103251. DOI: https://doi.org/10.1016/j.autcon.2020.103251.

Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. Automation in Construction, 92, 242-269. DOI: https://doi.org/10.1016/j.autcon.2018.03.023.

Zargar, S. H., & Alaghmandan, M. (2019). CORAL: introducing a fully computational plug-in for stadium design and optimization; a case study of finding optimal spectators' viewing angle. Architectural Science Review, 62(2), 160-170. DOI: https://doi.org/10.1080/00038628.2018.1559133.

Zhao, Y., & Mei, H. (2013). Dynamic simulation and analysis of daylighting factors for gymnasiums in mid-latitude China. Building and Environment, 63, 56-68. DOI: https://doi.org/10.1016/j.buildenv.2012.12.015.

Østergård, T., Jensen, R. L., & Maagaard, S. E. (2016). Building simulations supporting decision making in early design–A review. Renewable and Sustainable Energy Reviews, 61, 187-201. DOI: https://doi.org/10.1016/j.rser.2016.03.045.



JA: Journal Article | CP: Conference Paper

TOC

3 Design method development

This chapter proposes a Multi-Objective and Multi-Disciplinary Optimization (MOMDO) design method. The method is designed for the conceptual design of complex buildings such as indoor sports halls. It contains two subtypes that are respectively suitable for reducing existing design possibilities and sparking new design possibilities.

The chapter is structured as follows. First, it introduces the specific purpose and direction of the method development (Section 3.1). Then, it describes the phases and subtypes of the method (Section 3.2) and important computational techniques adopted by the method (Section 3.3). Finally, it concludes by summarizing the main research results and providing concluding remarks (Section 3.4).

Section 3.2-3.3 involves contents published in Journal Articles 1-2 (Yang et al., 2018; Yang et al., 2020).

3.1 Introduction

The specific purpose of the method development is to establish a new Multi-Objective and Multi-Disciplinary Optimization (MOMDO) design method that can incorporate dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) in a better manner.

The necessity of a new method has been shown in the literature review in Chapter 2. As stated in Section 2.4.3, the promising type of methods reviewed (e.g., Newton's and Kaushik and Janssen's methods that have incorporated dynamic and interactive re-formulation to varying degrees) still has room for improvement in terms of information and knowledge extraction. According to relevant literature, the aforementioned promising methods had not sufficiently discussed how to extract useful information

and knowledge for dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). Thus, a promising direction of the method development is to enhance the information and knowledge extraction for Optimization Problem Re-Formulation (Re-OPF), more precisely, to properly arrange relevant actions associated with the information and knowledge extraction. Furthermore, to support those actions, computational techniques are necessary. The computational support includes the techniques, directly and indirectly, useful for information and knowledge extraction.

3.2 The Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method

The proposed method is developed based on the aforementioned development direction. This section first describes the phases and subtypes of the method in a general way (Section 3.2.1); and then, it provides more specifics of the phases and subtypes respectively (Section 3.2.2 and 3.2.3).

3.2.1 The phases and subtypes of the method

The proposed method consists of three phases and contains two subtypes, as illustrated in FIG.3.1.

The three phases are:

- **Phase-I:** Optimization Problem Initial-Formulation (Initial-OPF), which is an ideation phase responsible for “formulating” an initial Multi-Objective Optimization (MOO) problem.
- **Phase-II:** Optimization Problem Re-Formulation (Re-OPF), which is an exploration phase responsible for “re-formulating” previous Multi-Objective Optimization (MOO) problems.
- **Phase-III:** Optimization Problem Solving (OPS), which is an optimization phase responsible for “solving” a final Multi-Objective Optimization (MOO) problem.

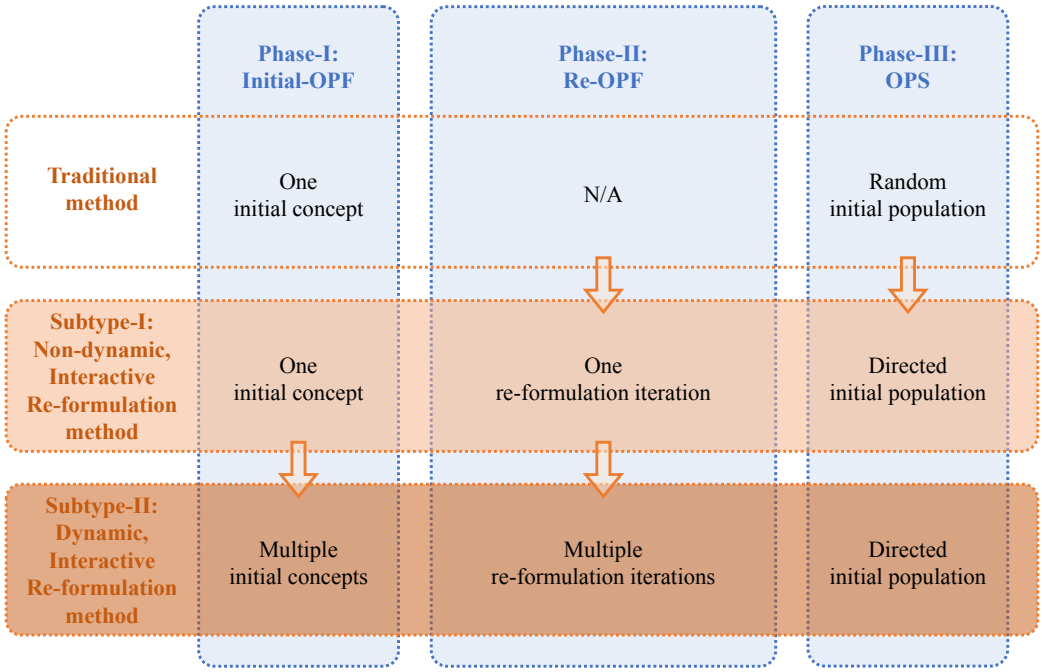


FIG. 3.1 The phases and subtypes of the proposed method

Note: The three blue columns represent the three phases of the proposed method; the first orange row represents a traditional method; the second and third orange rows represent the two subtypes of the proposed method; the arrows indicate where changes are made.

Among these phases, the re-formulation phase is the key one. This phase makes it possible to shift objective space and design space to include unexplored areas and/or exclude existing areas, thus facilitating the achievement of a more reliable optimization problem and more reliable optimal solutions. The incorporation of this phase is the main innovation of the proposed method, which differentiates this method from traditional methods (that do not incorporate the re-formulation phase).

The two subtypes are:

- **Subtype-I:** Non-dynamic, Interactive Re-formulation method, which is more suitable for the design context where the main purpose is to reduce existing design possibilities (i.e., shrink exploration space).

- **Subtype-II:** Dynamic, Interactive Re-formulation method, which is more suitable for the design context where the main purpose is to spark new design possibilities (i.e., expand exploration space).

Either of the two subtypes consists of the aforementioned three phases. As indicated by the names, these two subtypes are distinguished from each other mainly by their variations in the re-formulation phase. More precisely, they are different in the initial-formulation and re-formulation phases (i.e., the different numbers of initial concepts and of re-formulation iterations), but the same in the solving phase (i.e., the same way to get directed initial populations).

3.2.2 Specifics of the three phases

The three phases of the proposed method contain several groups of general actions that are appropriately arranged (i.e., Action A-G), as illustrated in FIG.3.2. These actions are described in this section, especially those in the key phase - the re-formulation phase (i.e., Action C-E).

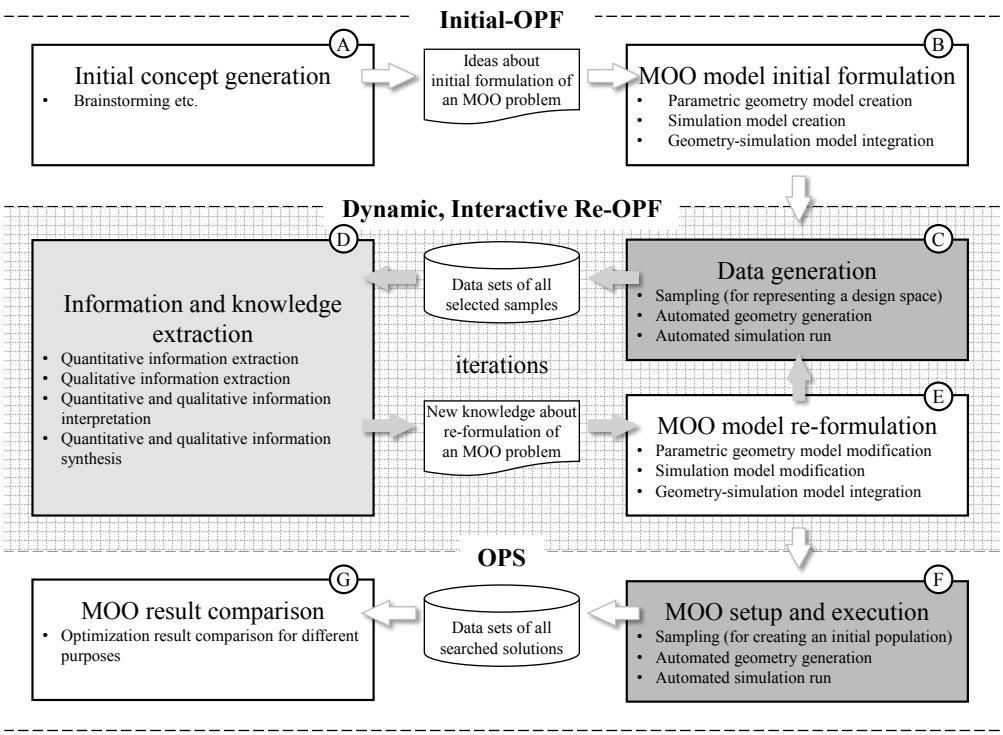


FIG. 3.2 The general actions required in the proposed method (Yang et al., 2020)

Note: The dark gray boxes show computer actions; the white boxes show human actions; and the light gray boxes show actions related to human-computer interaction.

TOC

3.2.2.1 Phase-I: Optimization Problem Initial-Formulation

To establish the initial-formulation phase, actions for initial concept generation (i.e., Action A) and MOO model initial formulation (i.e., Action B) are needed.

In initial concept generation (i.e., Action A), designers brainstorm and come up with one or multiple initial design concepts which are proposed to fulfil initial design requirements. Based on these concepts and requirements, designers can decide on initial design variables and performance measures.

In MOO model initial formulation (i.e., Action B), an initial parametric geometry model and simulation models are created, based on the preliminary ideas about design variables and performance measures. Then, these models are integrated to achieve an initial MOO model consisting of an initial set of performance objectives, constraints, and design variables. The initial MOO model, as the output of this phase, will be used to generate data for analysis in the next phase.

3.2.2.2 Phase-II: Optimization Problem Re-Formulation

To establish the re-formulation phase, actions for data generation (i.e., Action C), information and knowledge extraction (i.e., Action D), and MOO model re-formulation (i.e., Action E) are needed.

In data generation (i.e., Action C), a large number of samples are selected from the design space defined by the initial (or latest) MOO model; and the selected samples form a representation of the entire design space. Here, a sample means a vector of design variable values in a design space. Based on the initial (or latest) MOO model, the geometries of the selected samples are generated and related simulations are run automatically. This automation is driven by a predetermined sequential order. Then, qualitative data sets (i.e., images showing building geometries) and quantitative data sets (i.e., input values defining building geometries and output values representing performance results) of all selected samples are collected.

In information and knowledge extraction (i.e., Action D), the quantitative data sets are analyzed by computers to extract information about quantitative performances; and the qualitative data sets are observed by humans to extract information about qualitative performances. Then, the two kinds of information are interpreted and synthesized, to acquire comprehensive new knowledge about which design variables

and performance measures should be added or removed, namely new knowledge about the re-formulation of the MOO problem.

In MOO model re-formulation (i.e., Action E), the initial (or latest) parametric geometry model and simulation models are re-defined based on the acquired new knowledge. Then, these models are integrated to achieve a re-formulated MOO model consisting of a re-defined set of performance objectives, constraints, and design variables.

At this point, designers can decide either to continue the re-formulation phase by iterating through the above three groups of general actions (i.e., Action C-E), or, to enter the solving phase. This decision depends on the designers' satisfaction level of the re-formulated MOO model, and/or, the project's timeframe. After completing one or multiple re-formulation iterations, a final MOO model consisting of a final set of performance objectives, constraints, and design variables is achieved. The final MOO model, as the output of this phase, will be used to run optimizations in the next phase.

3.2.2.3 Phase-III: Optimization Problem Solving

To establish the solving phase, actions for MOO setup and execution (i.e., Action F) and MOO result comparison (i.e., Action G) are needed.

In MOO setup and execution (i.e., Action F), a small number of samples are selected from the design space defined by the final MOO model (more specifically from the high-performing clusters of samples in that design space); and the selected samples form a directed initial population for optimization. Based on the final MOO model, the geometries of searched solutions are generated and related simulations are run automatically. This automation is driven by an optimization algorithm. Then, qualitative and quantitative data sets of all searched solutions (i.e., optimal and non-optimal solutions) are collected.

In MOO result comparison (i.e., Action G), the qualitative and quantitative data sets of the optimal solutions are compared in order to extract relevant information and knowledge (e.g., trade-off relations between pairs of performance objectives, relative advantages of a particular optimal-design method over others), so as to support final design decision-making and/or optimal-design method verification.

3.2.2.4 The key phase

Among the above three phases, the re-formulation phase is the key one. This is because such re-formulation makes it possible to shift objective space and design space to include unexplored areas and/or exclude existing areas, thus enabling the achievement of a more reliable optimization problem and more realistic optimal solutions, as explained in Section 2.3.3. Three groups of general actions are needed to establish this phase (i.e., Action C-E), as described below.

Data generation (i.e., Action C) is the basis of the re-formulation phase. It is a process to provide design variable values and performance values for further analysis. In this process, advanced sampling algorithms are used to select samples, rather than pure random sampling algorithms or optimization algorithms. By doing so, the selected samples can become more representative (i.e., representing an entire design space rather than a small portion of it), thus helping to understand the overall performance trends of the entire design space; and the sampling can become more efficient (i.e., requiring relatively less samples to represent the design space), hence helping to reduce simulation time. Nevertheless, it is also possible to use optimization algorithms to select samples, when the divergence of exploration is not so highlighted.

Information and knowledge extraction (i.e., Action D) is the core of the re-formulation phase. It is essentially a process to transfer raw data into useful information and knowledge based on human-computer interaction, as illustrated in FIG.3.3. In this process, on one hand, advanced quantitative data analysis techniques (i.e., computational supports) are used to extract quantitative information, and such information is then interpreted by designers to form quantitative knowledge; on the other hand, human observations and judgments (i.e., human subjectivity) are used to extract qualitative information, and such information is then interpreted by designers to form qualitative knowledge; finally, these two categories of knowledge are synthesized into comprehensive new knowledge. By doing so, designers can gradually improve their understanding of performance objectives and constraints, design variables, and their interplay.

MOO model re-formulation (i.e., Action E) is the result of the re-formulation phase. It is a process to re-define previous parametric geometry models and simulation models based on the extracted knowledge, so as to obtain a new MOO model. In this process, designers are facilitated by flexible modeling techniques, especially when there are many performance measures and design variables to be added or removed.

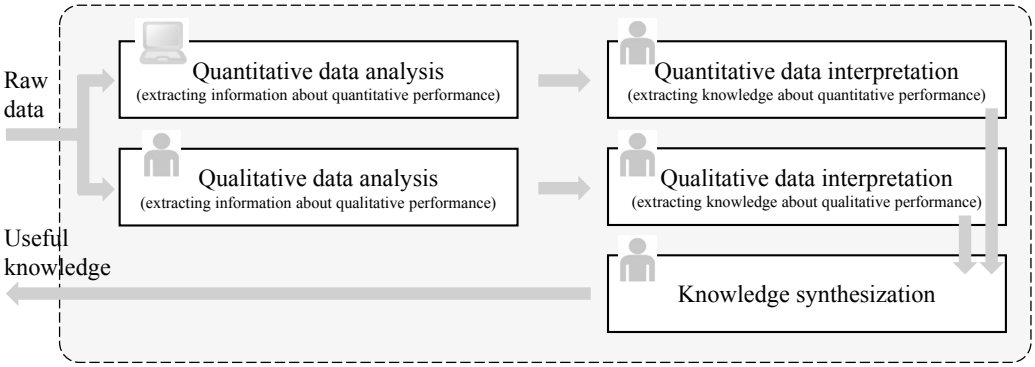


FIG. 3.3 Information and knowledge extraction relying on human-computer interaction

3.2.3 Specifics of the two subtypes

The two subtypes of the proposed method are distinguished from each other mainly by their different numbers of re-formulation iterations. The Subtype-I method (i.e., non-dynamic method) includes one re-formulation iteration to handle mainly convergent exploration. The Subtype-II method (i.e., dynamic method) includes multiple re-formulation iterations to handle mainly divergent exploration. For either subtype, three groups of general actions (i.e., Action C-E) are followed in each re-formulation iteration; but importantly, these actions can contain customizeable specific actions and result in different outcomes in each re-formulation iteration.

To better understand the two subtypes and their suitable design contexts, it is valuable to exemplify them by focusing on the specific actions and the outcomes of the re-formulation phase, as described below.

3.2.3.1 Subtype-I: Non-dynamic, Interactive Re-formulation method

To better understand this subtype method, an example of it is provided (see FIG.3.4). This example includes one re-formulation iteration, and the specific actions of the iteration are marked by C1, D1, E1 in FIG.3.4. It has also been described in one of the author's previous publications (Yang et al., 2018).

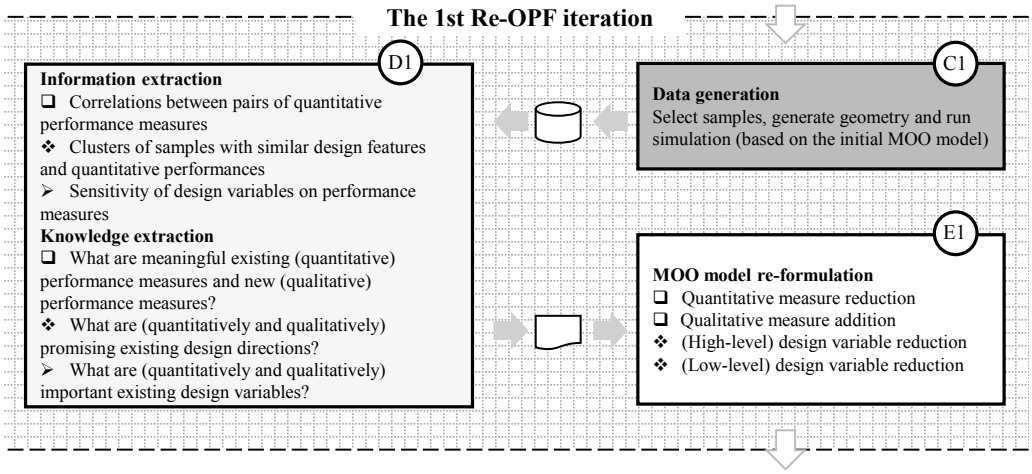


FIG. 3.4 The re-formulation phase in an example of the Subtype-I method

In the only re-formulation iteration, first, a data set is generated based on the initial MOO model. This MOO model may need to be adjusted (e.g., by adjusting design variable ranges and performance measure definitions) so that more feasible solutions and more properly defined measures can be obtained. Then, three kinds of quantitative information are extracted, thus aiding the following knowledge to be acquired: what are meaningful existing (quantitative) performance measures and new (qualitative) performance measures; what are (quantitatively and qualitatively) promising existing design directions; and what are (quantitatively and qualitatively) important existing design variables? The term “existing” is used to describe the ones which are already included in the model. Last of all, a re-formulated MOO model is obtained by conducting quantitative measure reduction, qualitative measure addition, and design variable reduction, as suggested by the extracted knowledge. This model is ready for the solving phase.

The above specific actions are just one possible option among many. Different specific actions can be conducted when using the Subtype-I method (i.e., non-dynamic method), which reflects the flexibility of this method. For instance, it is possible for designers to: extract a different combination of specific information and knowledge; and conduct mainly but not necessarily only convergent re-formulation.

3.2.3.2 Subtype-II: Dynamic, Interactive Re-formulation method

To better understand this subtype method, another example of it is provided (see FIG.3.5). This example includes three re-formulation iterations, and the specific actions of the iterations are marked by C1-C3, D1-D3, E1-E3 in FIG.3.5. It has also been described in one of the author's previous publications (Yang et al., 2020).

In the first re-formulation iteration, the first data set is generated based on the initial MOO model. Then, three kinds of quantitative information are extracted, thus aiding the following knowledge to be acquired: what are meaningful existing (quantitative) performance measures and new (qualitative) performance measures; what are (quantitatively and qualitatively) promising concepts among the initial ones; and what are (quantitatively and qualitatively) promising concepts besides the initial ones? Lastly, the first re-formulated MOO model is obtained by conducting quantitative measure reduction, qualitative measure addition, convergent concept selection, and divergent concept generation, as suggested by the extracted knowledge. This model is ready for the next re-formulation iteration.

In the second re-formulation iteration, the second data set is generated based on the first re-formulated MOO model. Then, one kind of quantitative information is extracted, thus aiding the following knowledge to be acquired: what are (quantitatively and qualitatively) promising concepts besides the existing ones? Lastly, the second re-formulated MOO model is obtained by conducting divergent concept generation, as suggested by the extracted knowledge. This model is ready for the next Re-OPF iteration.

In the third re-formulation iteration, the third data set is generated based on the second re-formulated MOO model. Then, one kind of quantitative information is extracted, thus aiding the following knowledge to be acquired: what are (quantitatively and qualitatively) promising concepts among all explored ones? Last of all, the third re-formulated MOO model is obtained by conducting convergent concept selection, as suggested by the extracted knowledge. This model is ready for the solving phase.

The above specific actions are just one possible option of many. More specific actions can be conducted when using the Subtype-II method (i.e., dynamic method), as this subtype method has a higher level of flexibility than the Subtype-I method (i.e., non-dynamic method). For instance, it is possible for designers to: include more than one but not necessarily three re-formulation iterations; extract a different combination of specific information and knowledge in each iteration; and conduct mainly but not necessarily only divergent re-formulation.

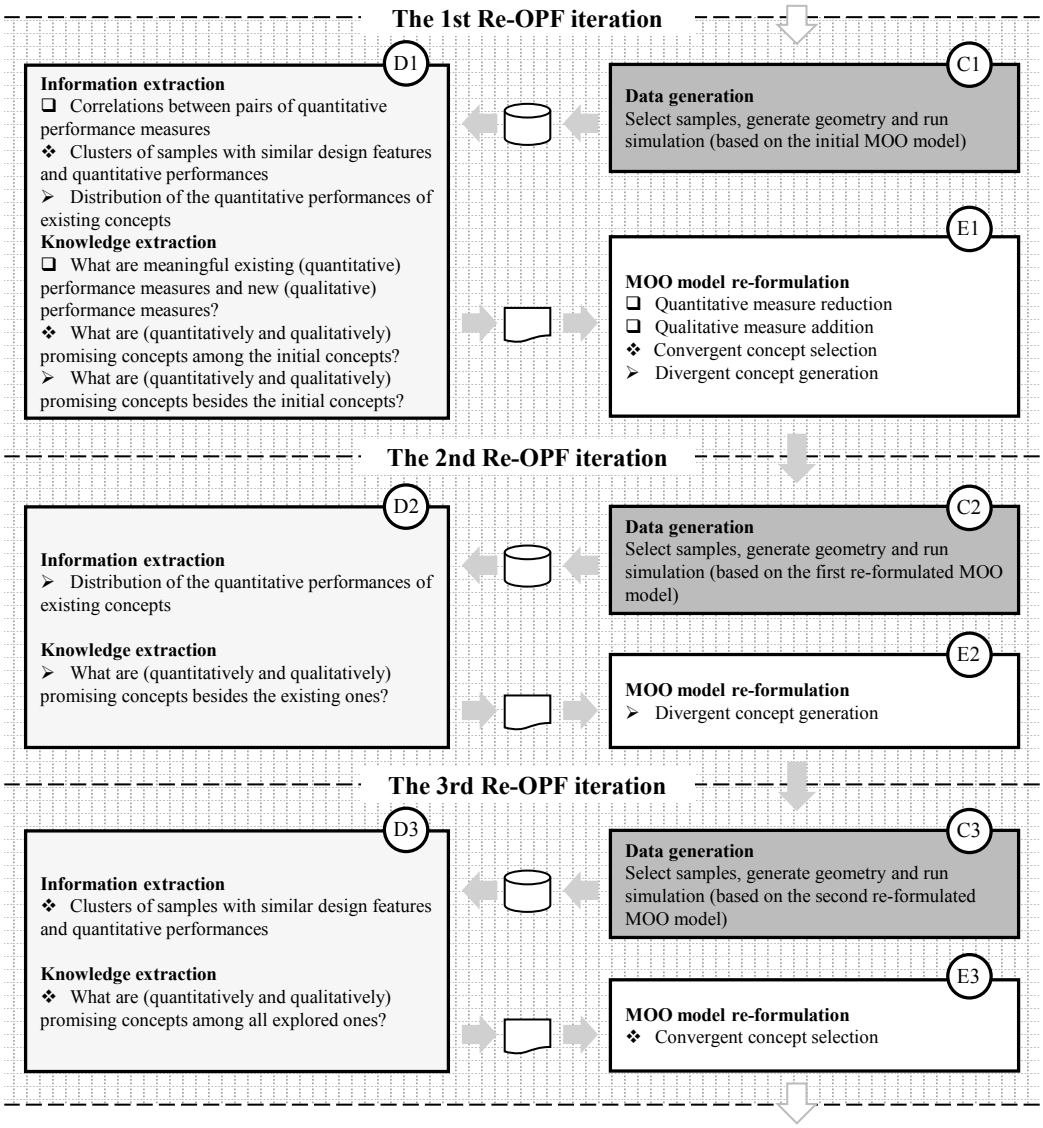


FIG. 3.5 The re-formulation phase in an example of the Subtype-II method (Yang et al., 2020)

3.2.3.3 Suitable design contexts

As shown above, the proposed method is in fact flexible due to the involvement of humans. The flexibility is especially reflected in the re-formulation phase where designers are allowed to customize specific actions. That is, designers can: include a desired number of re-formulation iterations and focus on different specific aspects of information and knowledge extraction (i.e., Action D) and MOO model re-formulation (i.e., Action E) in different iterations. To be more specific, they can: include one or multiple re-formulation iterations, and focus on quantitative or qualitative thinking, divergent or convergent thinking, to varying degrees, in different iterations.

The flexibility allows the proposed method to be applied in different design contexts. The relatively early sub-phase of conceptual architectural design is the right timing to spark new design possibilities, which can prevent overlooking valuable design variables (Liu et al., 2003). Concurrently, the relatively late sub-phase of conceptual architectural design is the good timing to reduce existing design possibilities, which can restrict the number of design variables from getting too large to allow meaningful considerations (Liu et al., 2003). Thus, the Subtype-I method (i.e., non-dynamic method), which handles mainly convergent exploration, is suitable for the latter design context; while the Subtype-II method (i.e., dynamic method), which handles mainly divergent exploration, is suitable for the former design context.

It is worth understanding that Pareto optimal solutions should have the following characteristics. From the perspective of quantitative performances, Pareto optimal solutions should be as close as possible to the true Pareto front, and as uniformly spread as possible within the decision maker's region of interest. That is, they should have good proximity, diversity, and pertinence (Fleming et al, 2005; Rostami and Shenfield, 2017). From the perspective of qualitative performances, Pareto optimal solutions should be as compliant as possible with the decision maker's subjective geometric preferences and have a proper degree of geometric variations. That is, they should be good in geometric preference compliance and geometric variation appropriateness.

Some of these characteristics can be understood differently in different design contexts. In the context that highlights reducing existing design possibilities, good pertinence means that Pareto solutions are within a small region of interest, good geometric variation appropriateness means that Pareto solutions have a low degree of geometric variations. In contrast, in the context that highlights sparking new design possibilities, good pertinence means that Pareto solutions are within a large region of interest, good geometric variation appropriateness means that Pareto solutions have a high degree of geometric variations.

3.3 Computational techniques

Given the importance of computational techniques, this section particularly focuses on presenting those unfamiliar to architects. More precisely, the computational techniques presented in this section are those that can directly and indirectly support the information and knowledge extraction during the Optimization Problem Re-Formulation (Re-OPF) phase.

As shown in Appendix IV, those techniques are grouped into the following three categories: techniques for data generation (Section 3.3.1), techniques for information and knowledge extraction (Section 3.3.2), and techniques for MOO model re-formulation (Section 3.3.3). Among these techniques, those for information and knowledge extraction are the key and especially highlighted in this research, while the others are not investigated in equal depth.

3.3.1 Techniques for data generation

3.3.1.1 Necessity of advanced sampling algorithms

Advanced sampling algorithms here refer to the algorithms that can be used to select samples in efficient ways. They are needed mainly because they are useful for the data generation during the Optimization Problem Re-Formulation (Re-OPF) phase, as briefly mentioned in Section 3.2.2.4. More specifically, they have relative advantages over other means of selecting samples, as described below.

In a traditional optimal-design method that involves no Optimization Problem Re-Formulation (Re-OPF), a pure random sampling algorithm and an optimization algorithm are often used in combination during the Optimization Problem Solving (OPS) phase. The former algorithm is used to select a small number of samples to serve as the initial population for optimization; while the latter algorithm is used to select the rest generations of samples that gradually approach optimization goals.

However, in an optimal-design method that involves Optimization Problem Re-Formulation (Re-OPF), more samples need to be preselected to run simulations. When using a pure random sampling algorithm together with an optimization

algorithm, it can lead to less representative samples (that only represent a small portion of the design space), which hinders the understanding of the overall performance trends of an entire design space. When using a pure random sampling algorithm alone, it can lead to less efficient sampling (that requires many samples to represent the design space), which increases simulation time. Thus, it is valuable to utilize advanced sampling algorithms to select representative samples efficiently for running simulations in the Optimization Problem Re-Formulation (Re-OPF) phase. Those algorithms can be seen as indirectly useful for the information and knowledge extraction, considering that the sample selection and simulation run are the basis of the information and knowledge extraction.

Design of experiments sampling algorithms (specifically Uniform Latin Hypercube Sampling) are the advanced sampling algorithms focused on in this research, as presented in the next section.

3.3.1.2 Design of experiments sampling: Uniform Latin Hypercube Sampling

Design of experiments sampling algorithms guide the choice of samples in a way that obtains the maximum amount of information using the minimum amounts of resources, namely using a lower number of samples (Cavazzuti, 2013). They can be used to select samples that can represent an entire design space, thus facilitating the exploration of the performance trends over the entire spectrum of that design space (Flager et al., 2009a). An advanced sampling algorithm differs from an optimization algorithm - the former selects all samples in one shot before running simulations, while the latter selects a small portion of samples at a time based on previous simulation results. An advanced sampling algorithm also differs from a purely random sampling algorithm - the former can use a relatively smaller number of samples than the latter, to represent the same design space.

Uniform Latin Hypercube (ULH) Sampling is a particular advanced sampling technique (McKay et al., 1979). Essentially, it is a stratified sampling technique. It splits the range of each input variable into N intervals of equal probability; then, each of the N partitions is sampled once (Davis et al., 2018). It can guarantee the lowest correlation between each pair of input variables and the highest uniform distribution (Clarich and Russo, 2011).

3.3.2 Techniques for information and knowledge extraction

3.3.2.1 Necessity of advanced quantitative data analysis techniques

Advanced quantitative data analysis techniques here refer to the techniques that can be used to analyze quantitative data in efficient ways. They are needed mainly because they are useful for the information and knowledge extraction during the Optimization Problem Re-Formulation (Re-OPF) phase, as briefly mentioned in Section 3.2.2.4. To be more specific, they have advantages in extracting various useful information, as described below.

In a traditional optimal-design method that involves no Optimization Problem Re-Formulation (Re-OPF), trade-off analysis is often used in the Optimization Problem Solving (OPS) phase. It is used to extract information about trade-off relations between pairs of performance objectives.

However, in an optimal-design method that involves Optimization Problem Re-Formulation (Re-OPF), more information needs to be extracted to aid with acquiring relevant knowledge. Such information includes but is not limited to correlations among performance measures, interplay between performance measures and design variables, sensitivity of performance measures to design variables. When still using trade-off analysis alone, it is hard to grasp all the above information. Thus, it is valuable to utilize advanced quantitative data analysis techniques to extract a broad range of information for acquiring relevant knowledge in the Optimization Problem Re-Formulation (Re-OPF) phase. Given this fact, those techniques can be seen as directly useful for the information and knowledge extraction.

In addition, it is worth noting that advanced quantitative data analysis techniques are also useful for the Optimization Problem Solving (OPS) phase. For instance, knowing the interplay between performance measures and design variables can help to obtain promising initial populations for optimization. Normally, an initial population is created by randomly selecting samples from an entire design space. In this way, the performances of the samples are not taken into account during the creation of the initial population. This can lead to unfavorable initial population and cause the search to start from low-performing samples, thus limiting the search efficiency. In fact, it would be useful to create an initial population by selecting samples from high-performing clusters of samples (but knowing these clusters requires advanced quantitative data analysis). In this way, the performances of the samples are an important concern during the creation of the initial population;

and the initial population created is called *a directed initial population*. This can bring about promising initial population and make the search starting from high-performing samples, thus improving the search efficiency.

Correlation analysis, cluster analysis, and sensitivity analysis (specifically Self-Organizing Map, Hierarchical Clustering, and Smoothing Spline Analysis of Variance) are the advanced quantitative data analysis techniques focused on in this research, as presented in the following three sections.

3.3.2.2 Correlation analysis: Self-Organizing Map

Correlation analysis measures the strength and direction of the relationship between two quantitative variables (Bobko, 2001; Chen and Popovich, 2002; Franzese and Iuliano, 2019). It can be used to extract correlations between pairs of quantitative performance measures. Knowing this information, can help to identify meaningful quantitative performance measures from possible ones. Specifically, when two measures are positively and strongly correlated and their optimization directions are the same, or, when two measures are negatively and strongly correlated and their optimization directions are opposite, there are probably no meaningful trade-off relations between the two measures. Thus, one of the measures can be considered meaningful and kept for further exploration, while the other one can be removed or treated as a constraint.

Self-Organizing Map (SOM) is a particular way of conducting correlation analysis (Kohonen, 2001). Essentially, it is an unsupervised neural network for ordering of high-dimensional data in such a way that similar data are grouped spatially close to one another (Di Stefano, 2009). Concisely, it is a dimensionality reduction method which maps multi-dimensional data into a two-dimensional space. It is suitable for hunting for correlations, given its intuitive way of visualizing and interpreting SOM planes (Vesanto, 1999; Vesanto and Ahola, 1999; Himberg et al., 2001; Köhler et al., 2010).

The principle of the Self-Organizing Map (SOM) technique is shown in FIG.3.6 and described below. First, a learning algorithm is applied to a training data set that has three dimensions (i.e., quantitative performance measures X, Y and Z), in order to generate prototype vectors (i.e., the vectors whose distribution approximates the probability density function of the training data set, as defined by Köhler et al.). The prototype vectors gradually approach the distribution of the training data set by using the algorithm; each prototype vector may correspond to a subset of the training data.

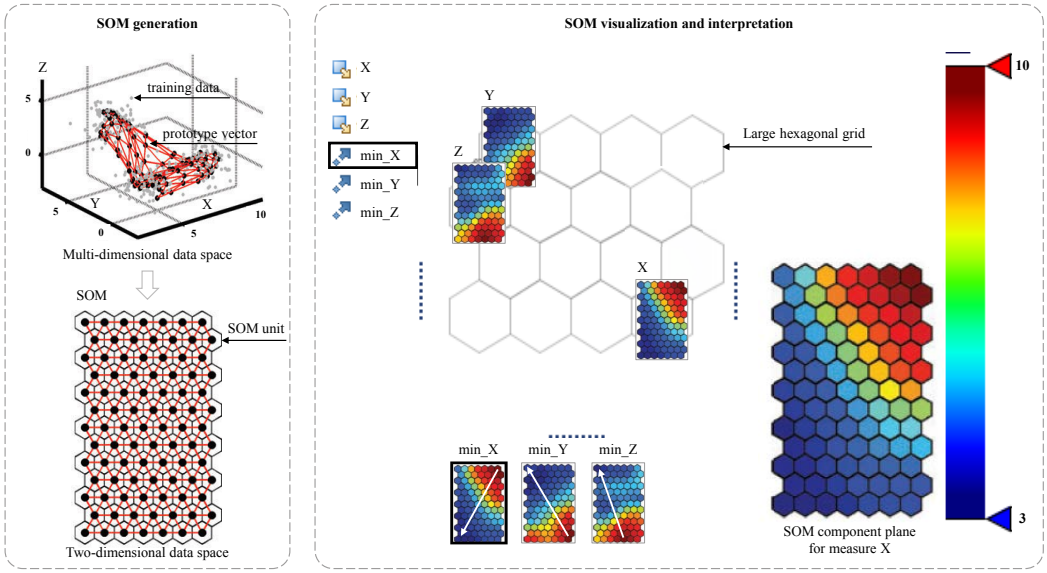


FIG. 3.6 Self-Organizing Map (revised from Köhler et al., 2010)

Note: SOM generation (left); SOM visualization and interpretation (right).

Then, the prototype vectors obtained are projected onto a two-dimensional honeycomb-like diagram (unfilled with colors). This diagram is the basis of generating a SOM plane (filled with colors) for each dimension. SOM plane's units are colored based on the values of the obtained prototype vectors. The color scale from blue to red represents the values from low to high. Finally, SOM planes for all dimensions are arranged on a large hexagonal grid, according to the correlations of the dimensions. The stronger the correlation between two dimensions, the more similar the SOM planes' color patterns and the closer the SOM planes' positions. Thus, by visually observing the SOM planes, the correlations between pairs of all dimensions can be quickly understood. For instance, the SOM planes for Y and Z have similar color patterns and are closer with each other, which means that Y and Z are strongly correlated; while the SOM planes for X and Y have very different color patterns and are far away from each other, which means that X and Y are weakly correlated. As for the correlation directions, when the same areas of the SOM planes have more similar colors, the associated dimensions are more likely to be positively correlated.

Arrows on top of the SOM planes can show the optimization directions. Those pointing from a low-value area to a high-value area (i.e., from blue to red) represent maximization goals; while those having the opposite direction represent minimization

goals. Thus, by simultaneously observing the SOM planes and the arrows, it is convenient to know the strength and direction of each pair of dimensions, and the optimization directions of these dimensions, so as to know if there are meaningful trade-off relations between the dimensions in question.

3.3.2.3 Cluster analysis: Hierarchical Clustering

Cluster analysis identifies homogeneous clusters of samples in a source data set based on measured characteristics (Di Stefano, 2009). It can be used to extract clusters of samples with similar design features and quantitative performances. Knowing this information, can help to identify quantitatively promising design concepts (or directions) from existing ones. Specifically, quantitatively high-performing clusters can be identified from all clusters, by narrowing down focused performance ranges towards optimization directions to desired extents; when the clusters identified mostly belong to a design concept, this concept is reasonably believed to be more competitive. Thus, this concept can be considered quantitatively promising and kept for further exploration.

Hierarchical Clustering (HC) is a particular way of conducting cluster analysis (Di Stefano, 2009). Essentially, it is a versatile data clustering approach which produces a nested series of partitions rather than only one partition (Jain et al., 1999). Concisely, it is a data structure refinement method which can provide refined views to the inherent structure of the data. It is suitable for creating clusters, given its flexible way of grouping a large amount of data into manageable and meaningful clusters.

The principle of the Hierarchical Clustering technique is shown in FIG.3.7 and described below. First, a clustering algorithm is applied to a source data set that has four dimensions (i.e., quantitative performance measures X, Y and Z, and a design variable called Concept), in order to generate desired clusters. Similar samples in the source data set are gradually merged by using the algorithm; larger clusters created at later stages are based on smaller clusters created at earlier stages, thus forming nested clusters. Then, the nested clusters are represented by a tree-like diagram called dendrogram. By choosing the position of a horizontal dash line that intersects with the dendrogram, the number of clusters to be applied to the source data set are determined (each intersection represents a cluster). Finally, the clusters applied are visualized using a clustering parallel coordinate chart. They are represented by different colored bands. For each colored band, the intersections of the center polyline and the parallel vertical lines show the mean values of the clusters at different dimensions; and the band width shows the confidence intervals of the mean values.

Thus, by visually observing the colored bands, the relative trends of the design concepts (or directions) and their performance values can be quickly understood. For instance, the design concept (or direction) number two has relatively high X values, relatively low Y values and medium Z values.

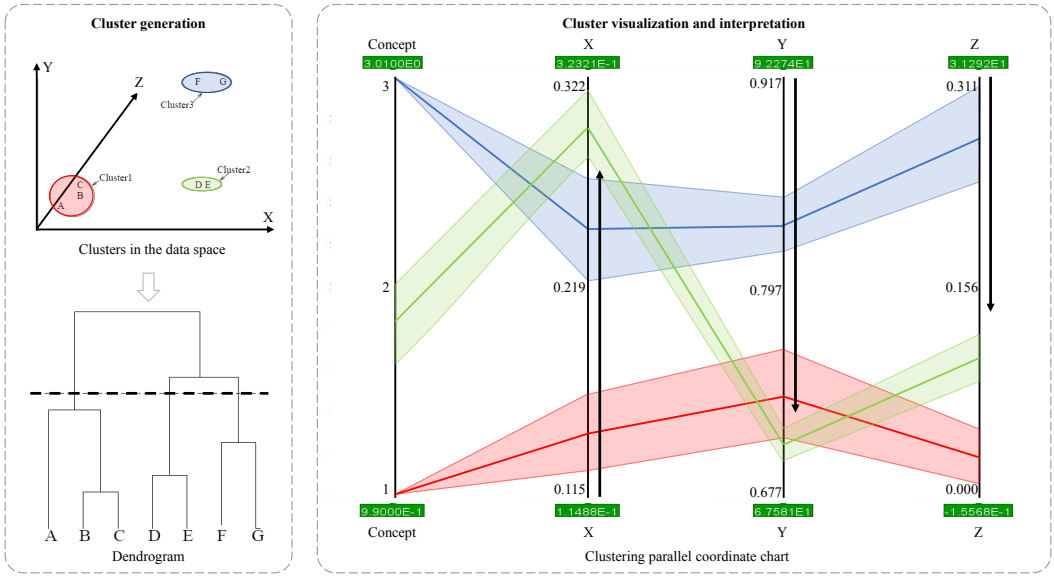


FIG. 3.7 Hierarchical Clustering (revised from Jain et al., 1999)

Note: Cluster generation (left); cluster visualization and interpretation (right).

Arrows along with the parallel vertical lines can show the optimization directions. Those pointing from a low-value to a high-value represent maximization goals while those having the opposite direction represent minimization goals. Thus, by collectively observing the colored bands and the arrows, it is convenient to know the clusters that best conform to the optimization directions of all dimensions, so as to know quantitatively promising design concepts (or directions).

3.3.2.4 Sensitivity analysis: Smoothing Spline Analysis of Variance

Sensitivity analysis estimates how the variations of the input variables of a mathematical model influence the response values (Tzuc et al., 2019). It can be used to extract sensitivity of design variables on performance measures. Knowing this information, can help to identify important design variables that contribute the most to the variation of performance measures. Specifically, all design variables are ranked in the order of sensitivity (or importance); then, high-ranked design variables which account for the major portion of a cumulative effect are considered important and kept for further exploration.

Smoothing Spline Analysis of Variance (SS-ANOVA) is a particular way of conducting sensitivity analysis (Gu, 2002). Essentially, it is a statistical modeling algorithm based on a function decomposition similar to the classical Analysis of Variance (ANOVA) decomposition and the associated notions of main effects and interaction; concisely, it is a function estimation method for both univariate and multivariate regression problems (Ricco et al., 2013). It differs from the classical ANOVA, mainly in that it is a more flexible nonparametric regression model that can vary in a high-dimensional function space, rather than a classical parametric regression model that has a given fixed form (Touzani and Busby, 2013). It is suitable for screening design variables, given this flexibility.

3.3.3 Techniques for multi-objective optimization model re-formulation

3.3.3.1 Necessity of flexible modeling techniques

Flexible modeling techniques here refer to the techniques that can be used to create MOO models in flexible ways. They are needed mainly because they are useful for the MOO model re-formulation during the Optimization Problem Re-Formulation (Re-OPF) phase, as briefly mentioned in Section 3.2.2.4. To be more specific, they have advantages in modifying parametric geometric models and multi-disciplinary simulation models, as described below.

In a traditional optimal-design method that involves no Optimization Problem Re-Formulation (Re-OPF), modeling techniques with limited flexibility are often used in the Optimization Problem Initial-Formulation (Initial-OPF) phase (Davis et al., 2011).

They are used to build parametric geometric models and multi-disciplinary simulation models in one shot without the need to modify them later.

However, in an optimal-design method that involves Optimization Problem Re-Formulation (Re-OPF), parametric geometric models and multi-disciplinary simulation models need to be modified flexibly. When still using less flexible modeling techniques, it can lead to cumbersome model modification (Davis et al., 2011) or low conceptual variety (Kilian, 2006). Thus, it is valuable to utilize flexible modeling techniques for modifying parametric geometric models and multi-disciplinary simulation models in the Optimization Problem Re-Formulation (Re-OPF) phase. Those techniques can be seen as indirectly useful for the information and knowledge extraction, considering that the model modification is the basis of a new round of information and knowledge extraction.

In addition, it is worth noting that flexible modeling techniques are also useful for the Optimization Problem Initial-Formulation (Initial-OPF) phase. For instance, they can help to incorporate broader initial design concepts and performance criteria. Normally, only one initial concept and a limited number of criteria are considered in the initial MOO model. This is fine when aiming to reduce existing design possibilities and when the performance requirements are few. But, in contrast, when aiming to spark new design possibilities and the performance requirements are many, it is meaningful to consider broader initial concepts and criteria in the initial MOO model. This can increase the variety of concepts and criteria, thus helping to find good solutions.

Hierarchical variable structure and Modular programming (specifically Two-Level Variable Structure, Geometry and Simulation Modular Programming) are the flexible modeling techniques focused on in this research, as presented in the following two sections.

3.3.3.2 Hierarchical variable structure: Two-Level Variable Structure

Hierarchical variable structure facilitates the inclusion of different sets of design variables during parametric geometric modeling. It exists in a product design in which a number of substructures and parts are hierarchically assembled into a larger system (Yoshimura and Izui, 2002). Design variables in such product designs may be from different levels of the hierarchy; thus naturally, they are organized in a hierarchical variable structure, rather than a flat, one-dimensional array structure. This hierarchical structure consists of high-level and low-level variables. The value

of a high-level variable determines the selection of low-level variables. A low-level variable can correspond to one or multiple values of a high-level variable or does not correspond to any values of a high-level variable. Thus, the dimensionality of the design space defined by the high-level and low-level variables is changeable.

Two-Level Variable Structure is a particular hierarchical variable structure. It is useful, especially when there are many design concepts (or directions) to be defined parametrically. In the example shown in FIG.3.8 (top left), the high-level variable (i.e., input variable 0 called "Concept") is used to define the type of design concepts (or directions) to be investigated; and the low-level variables (i.e., the rest input variables) are used to define the geometries related to the design concepts (or directions). Python scripting is used to realize this two-level variable structure. As shown by the scripting in FIG.3.8 (bottom), the high-level variable controls the switch from one design concept (or direction) to another; that is, when the value of the high-level variable is determined, a subset of low-level design variables is chosen from the full variable set, in order to parametrically define the geometries of the design concept (or direction) that corresponds to the value of the high-level variable.

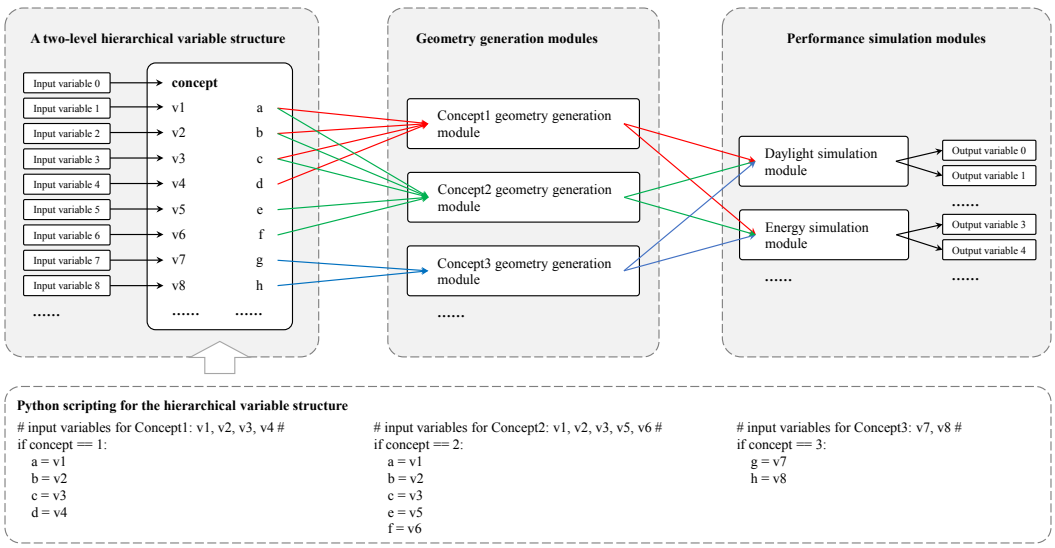


FIG. 3.8 The overall structure of parametric schemata

Note: 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).

3.3.3.3 Modular programming: Geometry and Simulation Modular Programming

Modular programming facilitates the modification of parametric schemata. It structures parametric schemata into different modules. A module in a dataflow programming language is defined as a sequence of program instructions bounded by entry and exit points; and it performs a particular task (Wong and Sharp, 1992). The entry points collect data the module requires; the exit points return data the module produces; and the program instructions in between can be evoked by passing data through the module. Parametric schemata structured using the modular programming principle are considered consistently better understood, especially when the parametric model is complex or used in a multi-disciplinary collaborative design environment (Davis et al., 2011).

Geometry and Simulation Modular Programming are respectively used to create geometry generation modules and performance simulation modules. In the example shown in FIG.3.8 (top middle and top right), each geometry module corresponds to a group of program instructions for defining a particular design concept (or direction); and each simulation module corresponds to a group of program instructions for defining a particular type of performance simulations.

3.4 Conclusion

This chapter concludes by summarizing the main research results (Section 3.4.1), and providing concluding remarks (Section 3.4.2).

3.4.1 Main research results

The main research results of this chapter include the following:

- The direction of the method development has been identified, which is to enhance the information and knowledge extraction for Optimization Problem Re-Formulation (Re-OPF), more precisely, to properly arrange relevant actions associated with the information and knowledge extraction. To support those actions, computational support needs to be provided.

- The three phases of the proposed method have been specified: Phase-I: Optimization Problem Initial-Formulation (Initial-OPF); Phase-II: Optimization Problem Re-Formulation (Re-OPF); and Phase-III: Optimization Problem Solving (OPS). Among these phases, the re-formulation phase is the key one. The incorporation of the re-formulation phase is the main innovation which differentiates the proposed method from traditional methods (that do not incorporate the re-formulation phase).

- The two subtypes of the proposed method have been specified: Subtype-I: Non-dynamic, Interactive Re-formulation method; and Subtype-II: Dynamic, Interactive Re-formulation method. They both consist of the aforementioned three phases. They are distinguished from each other mainly by their variations in the re-formulation phase. Their flexibility allows the proposed method to be applied in different design contexts, including the context that highlights reducing existing design possibilities and the context that highlights sparking new design possibilities.

- Some important computational techniques adopted in the Optimization Problem Re-Formulation (Re-OPF) phase have been specified. They include but are not limited to: Uniform Latin Hypercube Sampling for data generation; Self-Organizing Map, Hierarchical Clustering, Smoothing Spline Analysis of Variance for information and knowledge extraction; Two-Level Variable Structure, Geometry and Simulation Modular Programming for MOO model re-formulation.

3.4.2 Concluding remarks

In conclusion, the proposed method highlights the importance of dynamic and interactive Optimization Problem Re-Formulation (Re-OPF), specifically, the importance of the techniques directly and indirectly useful for information and knowledge extraction. The knowledge-supported re-formulation makes it possible to shift objective space and design space to include unexplored areas and/or exclude existing areas, which is beneficial for achieving a more reliable optimization problem, and obtaining reliable design solutions. The proposed method, including the two subtypes, will be applied to two case studies of the conceptual design of indoor sports halls in Chapter 5 and 6.

References

Bobko, P. (2001). Correlation and regression: Applications for industrial organizational psychology and management. Sage Publications, Thousand Oaks. ISBN: 978-0-7619-2303-9.

Cavazutti, M. (2013). Optimization Methods: From Theory to Design. Springer, Heidelberg, New York, Dordrecht, London. ISBN: 978-3-642-31186-4.

Chen, P. Y., & Popovich, P. M. (2002). Correlation: Parametric and nonparametric measures (No. 139). Sage Publications, Thousand Oaks. ISBN: 978-0761922285.

Clarich, A., & Russo, R. (2011). 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 (2011-006). ECCOMAS. ISBN: 978-605-61427-4-1.

Davis, D., Burry, B., & Burry, M. (2011). 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 (pp. 55–68). CAAD Futures and University of Liege. ISBN: 978-2-8745-6142-9.

Davis, D., Burry, J., & Burry, M. (2011). The flexibility of logic programming: Parametrically regenerating the Sagrada Família. In: C. M. 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 (pp. 29-38). CAADRIA and University of Newcastle. ISBN: 978-988-19026-2-7

Davis, S. E., Cremaschi, S., & Eden, M. R. (2018). Efficient surrogate model development: Impact of sample size and underlying model dimensions. Computer Aided Chemical Engineering, 44, 979-984. DOI: https://doi.org/10.1016/B978-0-444-64241-7.50158-0.

Di Stefano, D. (2009). Multivariate analysis algorithms in modeFRONTIER (v4). Technical report, ESTECO. https://engineering.esteco.com/modefrontier/ [Accessed: 24-September-2021].

Flager, F., Welle, B., Bansal, P., Soremekun, G., & Haymaker, J. (2009a). Multidisciplinary process integration and design optimization of a classroom building. Journal of Information Technology in Construction (ITcon), 14(38), 595-612. http://www.itcon.org/2009/38 [Accessed: 24-September-2021].

Fleming, P.J., Purshouse, R.C., Lygoe, R.J. (2005). Many-objective optimization: An engineering design perspective. In: C. A. Coello Coello, A. H. Aguirre, & E. Zitzler (Eds.), Evolutionary Multi-Criteri-on Optimization: Third International Conference (pp. 14-32). Springer-Verlag, Berlin, Heidelberg. ISBN: 978-3-540-24983-2.

Franzese, M., & Iuliano, A. (2019). Correlation analysis. Encyclopedia of Bioinformatics and Computational Biology, 1, 706-721. DOI: https://doi.org/10.1016/B978-0-12-809633-8.20358-0.

Gu, C. (2013). Smoothing Spline ANOVA Models. Springer, New York. ISBN: 978-0-387-95353-3.

Himberg, J., Ahola, J., Alhoniemi, E., Vesanto, J., & Simula, O. (2001). The Self-Organizing Map as a tool in knowledge engineering. In: N. R. Pal (Ed.), Pattern Recognition in Soft Computing Paradigm (pp. 38-65). World Scientific, Singapore. ISBN: 978-981-02-4491-0.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323. DOI: https://doi.org/10.1145/331499.331504.

Kilian, A. (2006). Design exploration through bidirectional modeling of constraints (doctoral dissertation). Department of Architecture, Massachusetts Institute of Technology, USA.

Köhler, A., Ohrnberger, M., & Scherbaum, F. (2010). Unsupervised pattern recognition in continuous seismic wavefield records using self-organizing maps. Geophysical Journal International, 182(3), 1619-1630. DOI: https://doi.org/10.1111/j.1365-246X.2010.04709.x.

Kohonen, T. (2001). Self-Organizing Maps. Third edition. Springer, Berlin, Heidelberg. ISBN: 978-3-540-67921-9.

Liu, Y. C., Chakrabarti, A., & Bligh, T. (2003). Towards an 'ideal' approach for concept generation. Design studies, 24(4), 341-355. DOI: https://doi.org/10.1016/S0142-694X(03)00003-6.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 42(1), 55-61. DOI: https://doi.org/10.2307/1268522.

Ricco, L., Rigoni, E., & Turco, A. (2013). Smoothing spline ANOVA for variable screening. Dolomites Research Notes on Approximation, 6 (Special_Issue), 130-139. DOI: https://doi.org/10.14658/pupj-drna-2013-Special_Issue-13.

Rostami, S., & Shenfield, A. (2017). A multi-tier adaptive grid algorithm for the evolutionary multi-objective optimisation of complex problems. Soft Computing, 21(17), 4963-4979. DOI: https://doi.org/10.1007/s00500-016-2227-6.

Touzani, S., & Busby, D. (2013). Smoothing spline analysis of variance approach for global sensitivity analysis of computer codes. Reliability Engineering & System Safety, 112, 67-81. DOI: https://doi.org/10.1016/j.ress.2012.11.008.

Turrin, M., Yang, D., D'Aquilio, A., Sileryte, R., & Sun, Y. (2016). Computational design for sport buildings. Procedia engineering, 147, 878-883. DOI: https://doi.org/10.1016/j.proeng.2016.06.285.

Tzuc, O. M., Bassam, A., Ricalde, L. J., & May, E. C. (2019). Sensitivity analysis with artificial neural networks for operation of photovoltaic systems. In: A. Y. Alanis, N. Arana-Daniel, & C. Lopez-Franco (Eds.), Artificial Neural Networks for Engineering Applications (pp. 127-138). Academic Press, Cambridge. ISBN: 978-0-12-818247-5.

Vesanto, J. (1999). SOM-based data visualization methods. Intelligent data analysis, 3(2), 111-126. DOI: https://doi.org/10.3233/IDA-1999-3203.

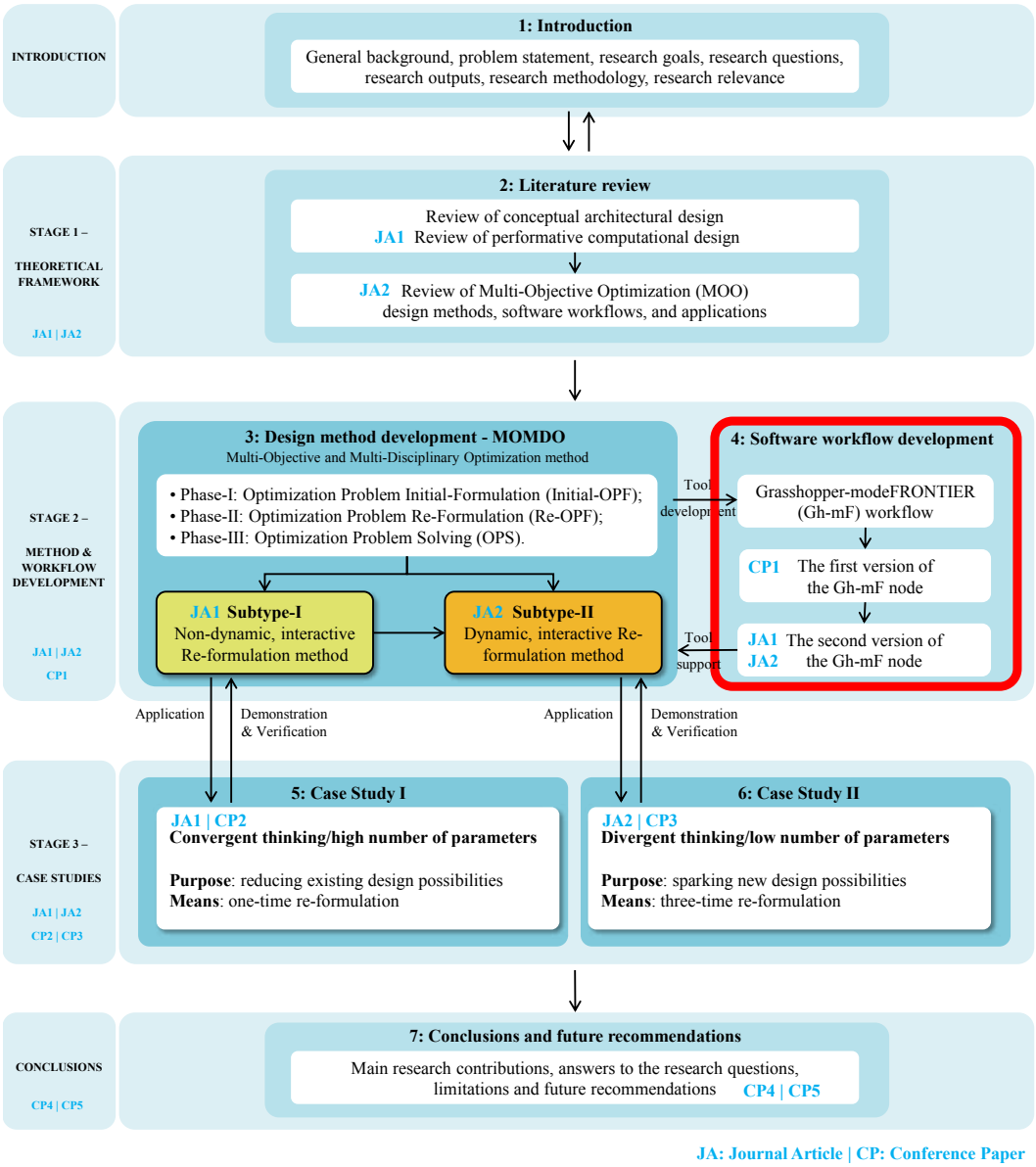
Vesanto, J., & Ahola, J. (1999). Hunting for correlations in data using the self-organizing map. In: H. Bother, E. Oja, E. Massad, C. Haefke (Eds.), Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications CIMA'99 (pp. 279-285). ICSC Academic Press. ISBN: 978-3-906454-18-4.

Wong, Y. K., & Sharp, J. A. (1992). A specification and design methodology based on data flow principles. In: J. A. Sharp (Ed.), Data Flow Computing: Theory and Practice (pp. 35-77). Ablex Publishing, New York. ISBN: 978-0-89391-654-1.

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, 103251. DOI: https://doi.org/10.1016/j.autcon.2020.103251.

Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. Automation in Construction, 92, 242-269. DOI: https://doi.org/10.1016/j.autcon.2018.03.023.

Yoshimura, M., & Izui, K. (2002). Smart optimization of machine systems using hierarchical genotype representations. J. Mech. Des., 124(3), 375-384. DOI: https://doi.org/10.1115/1.1486013.


INTRODUCTION
1: Introduction
General background, problem statement, research goals, research questions, research outputs, research methodology, research relevance
STAGE 1 – THEORETICAL FRAMEWORK
JA1 | JA2
2: Literature review
Review of conceptual architectural design
JA1 Review of performative computational design
JA2 Review of Multi-Objective Optimization (MOO) design methods, software workflows, and applications
STAGE 2 – METHOD & WORKFLOW DEVELOPMENT
JA1 | JA2
CP1
3: Design method development - MOMDO
Multi-Objective and Multi-Disciplinary Optimization method
• Phase-I: Optimization Problem Initial-Formulation (Initial-OPF);
• Phase-II: Optimization Problem Re-Formulation (Re-OPF);
• Phase-III: Optimization Problem Solving (OPS).
JA1 Subtype-I
Non-dynamic, interactive Re-formulation method
JA2 Subtype-II
Dynamic, interactive Re-formulation method
Tool development
Tool support
4: Software workflow development
Grasshopper-modeFRONTIER (Gh-mF) workflow
CP1 The first version of the Gh-mF node
JA1 JA2 The second version of the Gh-mF node
Application
Demonstration & Verification
Application
Demonstration & Verification
STAGE 3 – CASE STUDIES
JA1 | JA2
CP2 | CP3
5: Case Study I
JA1 | CP2
Convergent thinking/high number of parameters
Purpose: reducing existing design possibilities
Means: one-time re-formulation
6: Case Study II
JA2 | CP3
Divergent thinking/low number of parameters
Purpose: sparking new design possibilities
Means: three-time re-formulation
CONCLUSIONS
CP4 | CP5
7: Conclusions and future recommendations
Main research contributions, answers to the research questions, limitations and future recommendations CP4 | CP5
JA: Journal Article | CP: Conference Paper

TOC

4 Software workflow development

This chapter proposes a Multi-Objective and Multi-Disciplinary Optimization (MOMDO) software workflow. The workflow is designed for the implementation of the proposed method. It is realized by a newly developed integration node.

The chapter is structured as follows. First, it introduces the specific purpose and direction of the workflow development (Section 4.1). Then, it describes the basic software types to be integrated, and the selection and integration of specific software (Section 4.2). Finally, it concludes by summarizing the main research results and providing concluding remarks (Section 4.3).

Section 4.2 involves contents published in Journal Articles 1-2 (Yang et al., 2018; Yang et al., 2020) and Conference Paper 1 (Yang et al., 2015).

4.1 Introduction

The specific purpose of the workflow development is to establish an improved Multi-Objective and Multi-Disciplinary Optimization (MOMDO) software workflow where Grasshopper and modeFRONTIER are integrated in a better manner.

The necessity of an improved workflow has been shown in the literature review in Chapter 2. As stated in Section 2.5.3, the promising type of workflow reviewed (i.e., ESTECO's earliest in-house workflow that integrates Grasshopper and modeFRONTIER) still has room for improvement in terms of the software integration. According to the author's tests and the discussions with ESTECO's developers, the aforementioned promising workflow adopted a Grasshopper and modeFRONTIER integration plug-in that had limitations, such as tricky communication initiation, unstable automatic data exchange, and inefficient integration preparation. Thus,

a promising direction of the workflow development is to improve the integration between Grasshopper and modeFRONTIER, more precisely, to create a better integration plug-in that can improve the communication initiation, stabilize the automatic data exchange, and simplify the integration preparation. The workflow development is based on the collaboration between the Chair of Design Informatics at TU Delft and ESTECO SpA.

4.2 The Grasshopper-modeFRONTIER (Gh-mF) workflow

The proposed workflow is developed by integrating different types of software tools. This section first describes the basic software types to be integrated (Section 4.3.1); and then, it explains the selection of specific software tools (Section 4.3.2), and how the selected software tools are integrated (Section 4.3.3).

4.2.1 Basic software types to be integrated

According to Bernal et al. (2015), solution generation, solution evaluation, and solution selection software tools are relevant for early design stages; and they are worth being integrated by using a custom system-to-system approach. The software integration, or tool integration, not only refers to tool interoperability but also tool automation. The tool interoperability can enable multiple tools to communicate and cooperate with each other, regardless of differences in the implementation language, the execution environment, or the model abstraction (Madiajagan and Vijayakumar, 2008). The tool automation can automate data flows between interconnected tools, getting rid of the need to click icons and manually enter data to perform tasks (Díaz et al., 2017).

It is also worth integrating parametric modeling, performance simulation, and optimization software tools for early design stages, as these tools are specific examples of solution generation, solution evaluation, and solution selection software tools. Such integration can be further simplified as the integration of parametric modeling and optimization software tools because performance simulation software

tools may have already been embedded in those tools. The parametric modeling tool considered here is Visual Programming (VP) software (Boshernitsan and Downes, 2004); and the optimization tool considered here is Process Integration and Design Optimization (PIDO) software (Flager et al., 2009a).

The integration of Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software is needed in this research because it can offer necessary computational techniques for Multi-Objective Optimization (MOO) design methods, as briefly mentioned in Section 2.5.1. Without such integration, it is practically infeasible to conduct performance-based optimization, due to the involvement of many labor-intensive manual operations (e.g., manually generating geometries and inputting them for simulation runs, manually inputting performance results for optimization runs). In contrast, with such integration, it would be more convenient to conduct performance-based optimization, given the presence of automatic operations (e.g., automatic data manipulation and transfer). Moreover, such integration is needed also because it has the potential to provide computational techniques necessary for supporting Optimization Problem Re-Formulation (Re-OPF).

4.2.2 The selection of specific software

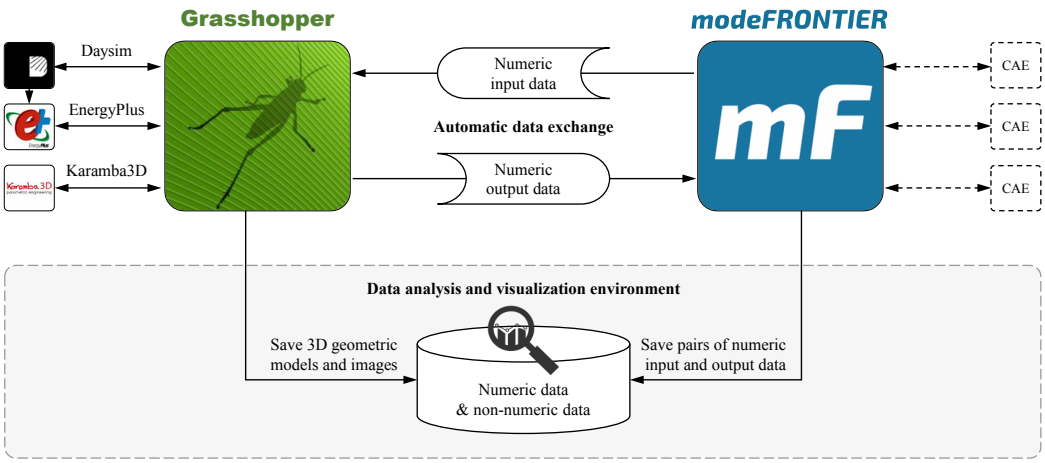


FIG. 4.1 The software tools involved in the proposed workflow (revised from Yang et al., 2020)

Note: Grasshopper and simulation software (top left); modeFRONTIER and simulation software (top right); and a database for storing numeric and non-numeric data (bottom).

Multiple specific software tools are selected to build the proposed workflow. As illustrated in FIG.4.1, these software tools include: McNeel's Grasshopper, ESTECO's modeFRONTIER, and simulation software Daysim, EnergyPlus, and Karamba3D (that are embedded in Grasshopper).

4.2.2.1 Choice of Grasshopper

McNeel's Grasshopper is a typical Visual Programming (VP) tool used for parametric design. It is the most prevalent parametric modeling tool among architectural design professionals, due to its intuitive way of exploring geometries without having to know scripting (Cichocka et al., 2017).

McNeel's Grasshopper has been chosen to build the proposed workflow, mainly because it can facilitate implementing Two-Level Variable Structure, Geometry and Simulation Modular Programming (mentioned in Section 3.3.3). To be more specific, Grasshopper can implement Two-Level Variable Structure by using Grasshopper's Python script editor and implement Geometry and Simulation Modular Programming by using Grasshopper's group and cluster features, as presented in Appendix IV.

It is also worth noting that Grasshopper is not the only possible choice for implementing the above techniques. Other possible choices include Bentley's GenerativeComponents, Autodesk's Dynamo Studio, and Sidefx' Houdini, etc.

4.2.2.2 Choice of modeFRONTIER

ESTECO's modeFRONTIER is a common Process Integration and Design Optimization (PIDO) tool used for multi-objective and multi-disciplinary engineering optimal-design. It is prevalent in many engineering industries, due to its capability of offering a seamless coupling with third-party engineering tools, enabling the automation of simulation processes, and facilitating analytic decision making etc.

ESTECO's modeFRONTIER has been chosen to build the proposed workflow, primarily given the fact that it can facilitate implementing Uniform Latin Hypercube Sampling, Self-Organizing Map, Hierarchical Clustering, and Smoothing Spline Analysis of Variance (mentioned in Section 3.3.1 and 3.3.2). To be more specific, modeFRONTIER can implement Uniform Latin Hypercube Sampling by using modeFRONTIER's design of experiments node, implement Self-Organizing Map

and Hierarchical Clustering by using modeFRONTIER's multivariate analysis tool, implement Smoothing Spline Analysis of Variance by using modeFRONTIER's sensitivity analysis tool, implement Five-Number Summary by using modeFRONTIER's distribution analysis chart, and implement Combined Data Visualization by using modeFRONTIER's run analysis interface, as presented in Appendix IV.

It is also worth noting that modeFRONTIER is not the only possible choice for implementing the above techniques. Other possible choices include Phoenix Integration's ModelCenter, Noesis Solutions' Optimus, and Dassault Systèmes' Isight, etc.

4.2.2.3 Choice of simulation software

Different kinds of simulation software are important tools for performance-based design. They are commonly used in architectural design, especially in a multi-disciplinary design environment.

Daylight simulation software Daysim, energy simulation software EnergyPlus, and structural simulation program Karamba3D have been chosen to build the proposed workflow, considering that they can facilitate implementing Integrated Dynamic Modeling (Negendahl, 2015) - they can be combined with Grasshopper. To be more specific, Daysim and EnergyPlus are combined with Grasshopper by using Grasshopper's plug-ins Ladybug and Honeybee; and Karamba3D itself is Grasshopper's plug-in.

It is also worth noting that the above kinds of simulation software are not the only possible choices. There can be various choices of third-party simulation software, which makes it possible to expand or customize the proposed workflow, and thus adapt to various needs in multi-disciplinary design environments. For instance, structural simulation software SAP2000 is another possible choice; it can be combined with Grasshopper by using Grasshopper's plug-in Geometry Gym; or it can also be combined with modeFRONTIER by using an Excel file (Rizzian et al., 2017).

4.2.3 The integration of specific software

The integration of Grasshopper and modeFRONTIER is the key to the proposed workflow. Such integration is achieved by using a newly developed integration plug-in called Grasshopper-modeFRONTIER (Gh-mF) node. The development of the new Gh-mF node starts with an existing prototype version. During the development process, at least two work-in-progress versions were produced. Based on these versions, the formal version has been finalized.

4.2.3.1 The prototype version of the integration node

The prototype version of Gh-mF node previously developed by ESTECO SpA aimed to preliminarily investigate the feasibility of integrating Grasshopper and modeFRONTIER. This version adopted a two-click approach to initiate the communication between Grasshopper and modeFRONTIER, due to some issues in using Grasshopper's API. According to the original integration scripts, first, by clicking a starting toggle from "*false*" to "*true*" in Grasshopper, Grasshopper starts to check for a message to be sent from modeFRONTIER every 10 milliseconds for 2000 times; thus, this gives the users a 20 second time slot to click a run icon in modeFRONTIER to initiate the communication. During the communication, Grasshopper is always kept alive to exchange data with modeFRONTIER automatically.

The prototype version of Gh-mF node has shown the feasibility to integrate Grasshopper and modeFRONTIER. However, it doesn't work all the time due to undetermined reasons. According to the author's tests, the following issues may be responsible for the sporadic malfunctions: it may fail to initiate the communication (which can be associated with the use of the two-click initiation approach); when the communication is occasionally initiated, unexpected connection errors may occur during the automatic data exchange (which can be associated with the way Grasshopper and modeFRONTIER communicate). Given the above limitations, new improved versions of Gh-mF node need to be developed. Unlike the prototype version that was created by using modeFRONTIER's EasyDriver feature, the new versions are expected to be created by using modeFRONTIER's more advanced myNODE feature (Duggan et al., 2012). The myNODE feature can pack integration scripts into a myNODE file that will be installed in modeFRONTIER. Once the myNODE file is installed, an integration plug-in is developed successfully.

4.2.3.2 The first version of the integration node

The first version of Gh-mF node developed in this research aims to overcome the limitations of the prototype version (i.e., tricky communication initiation and unstable automatic data exchange). This version adopts a one-click approach to initiate the communication between Grasshopper and modeFRONTIER, by taking advantage of Grasshopper's API. According to the new integration scripts, first, by clicking the run icon in modeFRONTIER, modeFRONTIER generates external input files, launches Grasshopper, and opens a Grasshopper file (where the starting toggle is always set to "*true*"); then, Grasshopper reads the external input files and generates external output files automatically; next, the Grasshopper file is closed and Grasshopper is shut down; last, modeFRONTIER collects the external output files, generates new external input files for driving the communication forward. During the communication, Grasshopper is launched and shut down repeatedly.

The first version of Gh-mF node manages to improve the communication initiation and stabilize the automatic data exchange (by avoiding the two-click initiation approach and changing the way Grasshopper and modeFRONTIER communicate). However, it still involves time-consuming manual operations in preparing the integration of Grasshopper and modeFRONTIER. For instance, the users have to manually create the templates of the external files in Grasshopper, manually specify variable names according to the templates in modeFRONTIER, and manually create Grasshopper components for saving 3D geometries. These manual operations make the integration preparation inefficient. Given the above limitation, a newer version of Gh-mF node needs to be developed. Unlike the first version that relies on an indirect communication approach (with the need of external files), the newer version is expected to rely on a direct communication approach (without the need of external files).

4.2.3.3 The second version of the integration node

The second version of Gh-mF node developed in this research aims to overcome the limitation of the first version (i.e., inefficient integration preparation). This version gets rid of manual operations to save time for the integration preparation, by taking advantage of Grasshopper's API. According to the newer integration scripts, Grasshopper and modeFRONTIER can directly communicate with each other; that is, the input and output variables involved in Grasshopper can be recognized and propagated to modeFRONTIER automatically; and 3D geometries generated in Grasshopper can be saved to modeFRONTIER automatically.

The second version of Gh-mF node manages to simplify the integration preparation (by getting rid of the manual operations). However, it is still a work-in-progress version, rather than a formally supported direct integration node by modeFRONTIER. Fine-tuning is needed to get the formal node.

In order to verify the aforementioned two work-in-progress versions, they have been tested in some simplified case studies (D'Aquilio et al., 2016; Sileryte et al., 2016).

4.2.3.4 The formal version of the integration node

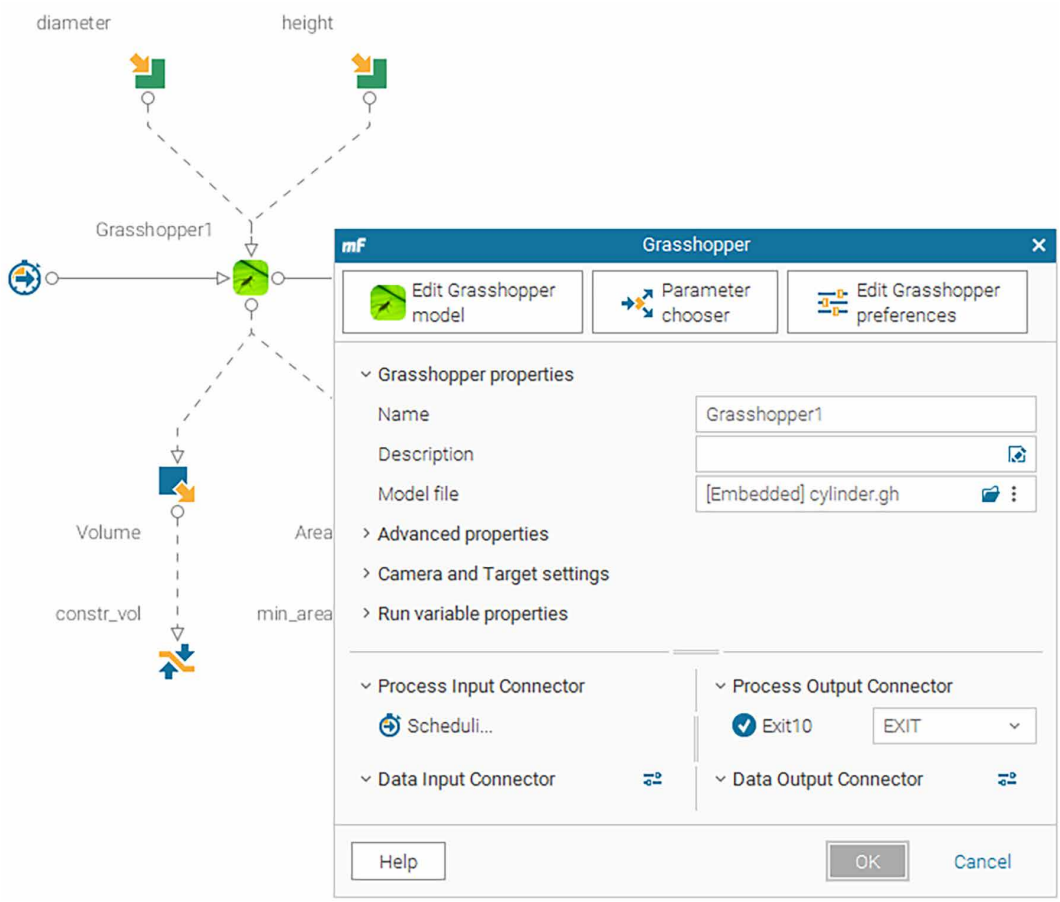


FIG. 4.2 The formal Gh-mF node available in modeFRONTIER (image from ESTECO's website https://www.esteco.com/corporate/volta-and-modefrontier-release-2020-winter-now-available)

TOC

Following several rounds of development and a final check, the formal version of Gh-mF node was finalized. It is a formally supported direct integration node available in modeFRONTIER, as illustrated in FIG.4.2.

The software workflow established by using the new Gh-mF node can automate the data exchange between Grasshopper and modeFRONTIER (see FIG.4.1). During the data exchange, Grasshopper is responsible for receiving numeric input data and generating geometries; the selected types of simulation software are responsible for receiving geometries and generating numeric output data; and modeFRONTIER is responsible for receiving numeric output data and generating new numeric input data for driving the automation forward. The interfaces of the automatic data exchange are shown in FIG.4.3. Moreover, the software workflow can also allow post-processing (i.e., analyzing and visualizing) the numeric and non-numeric data stored in a database. The interfaces of the post-processing are shown in FIG.4.4.

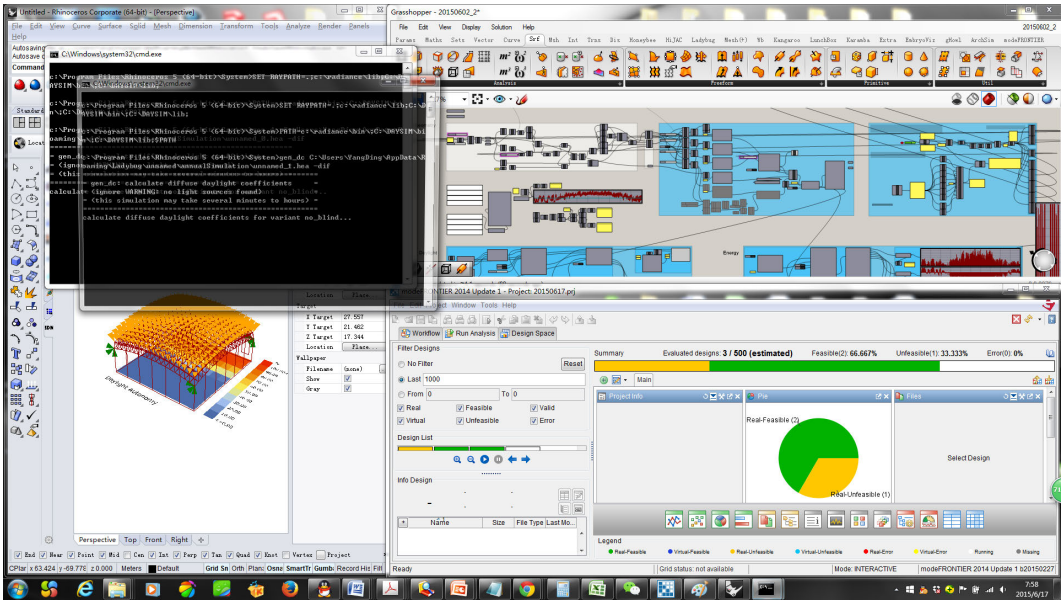


FIG. 4.3 Interfaces of the automatic data exchange (Yang et al., 2018)

TOC

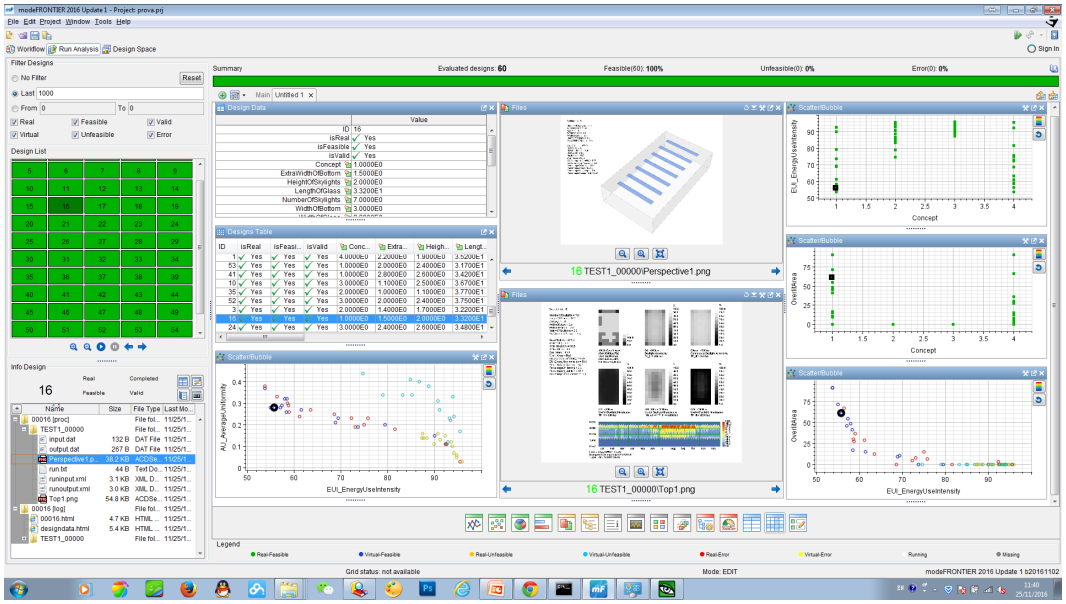


FIG. 4.4 Interfaces of the post-processing (Yang et al., 2018)

4.3 Conclusion

This chapter concludes by summarizing the main research results (Section 4.3.1) and providing concluding remarks (Section 4.3.2).

4.3.1 Main research results

The main research results of this chapter include the following:

- The direction of the workflow development has been identified, which is to improve the integration between Grasshopper and modeFRONTIER, more precisely, to create a better integration plug-in that can improve the communication initiation, stabilize the automatic data exchange, and simplify the integration preparation.

- The software tools involved in the proposed workflow have been specified: McNeel's Grasshopper, ESTECO's modeFRONTIER, and simulation software Daysim, EnergyPlus, and Karamba3D (that are embedded in Grasshopper). These software tools are selected for their capability of implementing the computational techniques adopted in the Optimization Problem Re-Formulation (Re-OPF) phase.
- The way of integrating Grasshopper and modeFRONTIER has been described. They are integrated by using a newly developed integration plug-in called Grasshopper-modeFRONTIER (Gh-mF) node. The new Gh-mF node is developed based on an existing prototype version; and it can improve the existing prototype version in terms of the communication initiation, automatic data exchange, and integration preparation.

4.3.2 Concluding remarks

In conclusion, the proposed workflow benefits from the integration of Grasshopper and modeFRONTIER, specifically, the integration achieved by using the newly developed integration plug-in. The new Gh-mF node can help implement the proposed method in a more reliable, stable and efficient manner, compared with the prototype version. The proposed workflow, established by using the new Gh-mF node, will be applied to two case studies of the conceptual design of indoor sports halls in Chapter 5 and 6.

References

Bernal, M., Haymaker, J. R., & Eastman, C. (2015). On the role of computational support for designers in action. Design Studies, 41, 163-182. DOI: https://doi.org/10.1016/j.destud.2015.08.001.

Boshernitsan, M., & Downes, M. S. (2004). Visual programming languages: a survey. Report of Computer Science Division, University of California, U.S. http://citeseerx.ist.psu.edu/viewdoc/download?-doi=10.1.1.73.2491&rep=rep1&type=pdf [Accessed: 24-September-2021].

Cichocka, J. M., Browne, W. N., & Rodriguez, E. (2017). Optimization in the architectural practice. 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 (pp. 387-396). CAADRIA and Xi'an Jiaotong-Liverpool University. ISBN: 978-988-19026-8-9.

D'Aquilio, A., Sileryte, R., Yang, D., & Turrin, M. (2016). Simulating natural ventilation in large sports buildings: prediction of temperature and airflow patterns in the early design stages. In: R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), 2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 43-50). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

Díaz, H., Alarcón, L. F., Mourgues, C., & García, S. (2017). Multidisciplinary Design Optimization through process integration in the AEC industry: Strategies and challenges. Automation in Construction, 73, 102-119. DOI: https://doi.org/10.1016/j.autcon.2016.09.007.

Duggan, A., Parashar, S., & Fateh, N. (2012). MyNode Concept NREC's TurboOPT modeFRONTIER's fast-track custom integration for Turbomachinary Optimization. Technical report, ESTECO. https://www.esteco.com/modefrontier/mynode-concept-nrecs-turboopt-modefrontier%E2%80%99s-fast-track-custom-integration-turbomachinary-optimization [Accessed: 24-September-2021].

Flager, F., Welle, B., Bansal, P., Soremekun, G., & Haymaker, J. (2009a). Multidisciplinary process integration and design optimization of a classroom building. Journal of Information Technology in Construction (ITcon), 14(38), 595-612. http://www.itcon.org/2009/38 [Accessed: 24-September-2021].

Madiajagan, M., & Vijayakumar, B. (2008). Interoperability in component based software development. International Journal of Computer and Systems Engineering, 2(10), 3507-3515. DOI: https://doi.org/10.5281/zenodo.1074609.

Negendahl, K. (2015). Building performance simulation in the early design stage: An introduction to integrated dynamic models. Automation in Construction, 54, 39-53. DOI: https://doi.org/10.1016/j.autcon.2015.03.002.

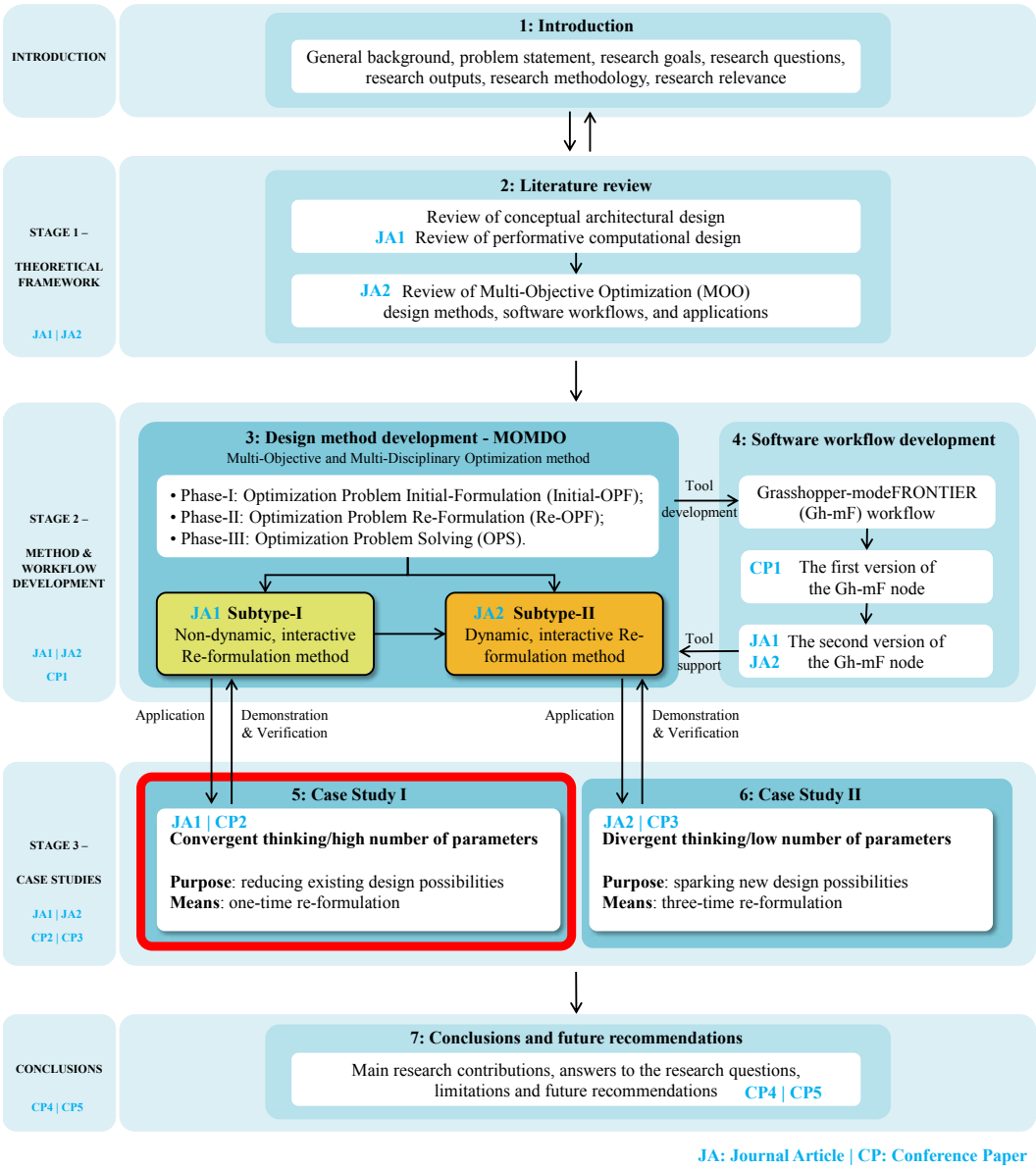
Rizzian, L., Léger, N., & Marchi, M. (2017). Multiobjective sizing optimization of seismic-isolated reinforced concrete structures. Procedia engineering, 199, 372-377. DOI: https://doi.org/10.1016/j.proeng.2017.09.044.

Sileryte, R., D'Aquilio, A., di Stefano, D., Yang, D., & Turrin, M. (2016). 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 (pp. 215-222). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

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, 103251. DOI: https://doi.org/10.1016/j.autcon.2020.103251.

Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. Automation in Construction, 92, 242-269. DOI: https://doi.org/10.1016/j.autcon.2018.03.023.

Yang, D., Turrin, M., Sariyildiz, S., & Sun, Y. (2015). Sports building envelope optimization using multi-objective multidisciplinary design optimization (M-MDO) techniques: Case of indoor sports building project in China. In: 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 2269-2278). IEEE, Piscataway. ISBN: 978-1-4799-7492-4.


INTRODUCTION
1: Introduction
General background, problem statement, research goals, research questions, research outputs, research methodology, research relevance
STAGE 1 – THEORETICAL FRAMEWORK
JA1 | JA2
2: Literature review
Review of conceptual architectural design
JA1 Review of performative computational design
JA2 Review of Multi-Objective Optimization (MOO) design methods, software workflows, and applications
STAGE 2 – METHOD & WORKFLOW DEVELOPMENT
JA1 | JA2
CP1
3: Design method development - MOMDO
Multi-Objective and Multi-Disciplinary Optimization method
• Phase-I: Optimization Problem Initial-Formulation (Initial-OPF);
• Phase-II: Optimization Problem Re-Formulation (Re-OPF);
• Phase-III: Optimization Problem Solving (OPS).
JA1 Subtype-I
Non-dynamic, interactive Re-formulation method
JA2 Subtype-II
Dynamic, interactive Re-formulation method
Tool development
4: Software workflow development
Grasshopper-modeFRONTIER (Gh-mF) workflow
CP1 The first version of the Gh-mF node
JA1 JA2 The second version of the Gh-mF node
Tool support
Application
Demonstration & Verification
Application
Demonstration & Verification
STAGE 3 – CASE STUDIES
JA1 | JA2
CP2 | CP3
5: Case Study I
JA1 | CP2
Convergent thinking/high number of parameters
Purpose: reducing existing design possibilities
Means: one-time re-formulation
6: Case Study II
JA2 | CP3
Divergent thinking/low number of parameters
Purpose: sparking new design possibilities
Means: three-time re-formulation
CONCLUSIONS
CP4 | CP5
7: Conclusions and future recommendations
Main research contributions, answers to the research questions, limitations and future recommendations CP4 | CP5
JA: Journal Article | CP: Conference Paper

TOC

5 Case Study I

This chapter presents Case Study I. In this case study, the Subtype-I method (i.e., non-dynamic method) is applied to the conceptual design of the overall geometry of a sports competition hall with the aid of the new Gh-mF node. This case study was selected to demonstrate and verify this subtype method primarily because it focused on the relatively late sub-phase of the conceptual design where convergent thinking is often highlighted or the number of parameters is usually high.

The chapter is structured as follows. First, it introduces the purpose of Case Study I (Section 5.1). Then, it provides the background of the project involved in this case study (Section 5.2). Next, it presents the results derived from each phase of the non-dynamic method (Section 5.3, 5.4, 5.5). Finally, it concludes by summarizing the main research results, identifying possible extensions of the non-dynamic method, and providing concluding remarks (Section 5.6).

This case study has collaborated with Arup Amsterdam. Ir. Shibo Ren, the senior structural engineer, helped to develop the structural model of the case and edit relevant texts. Sections 5.2-5.5 involve contents published in Journal Article 1 (Yang et al., 2018) and Conference Paper 2 (Yang et al., 2015).

5.1 Introduction

The purposes of Case Study I are multifold. First of all, this case study demonstrates the use of the Subtype-I method (i.e., non-dynamic method). Second, it verifies the benefits of adopting the method and the factors affecting the behaviors of the method. Third, it provides valuable feedback for possible extensions of the method.

This case study assumes that the design context is to highlight reducing existing design possibilities, such as many circumstances in the relatively late sub-phase of the conceptual design. Thus, the Subtype-I method (i.e., non-dynamic method) is adopted. This subtype method contains three phases; and the re-formulation phase is linear, as illustrated in FIG.5.1.

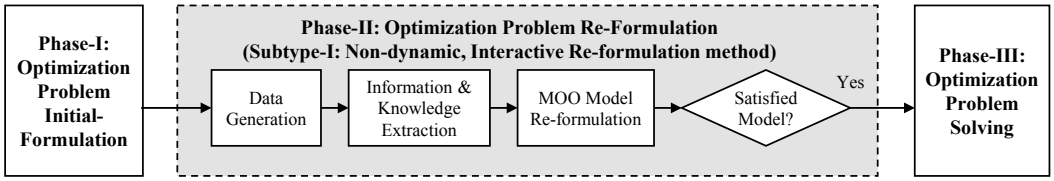


FIG. 5.1 The scheme of applying the Subtype-I method

Note: The shaded region corresponds to FIG.3.4.

5.2 Project description

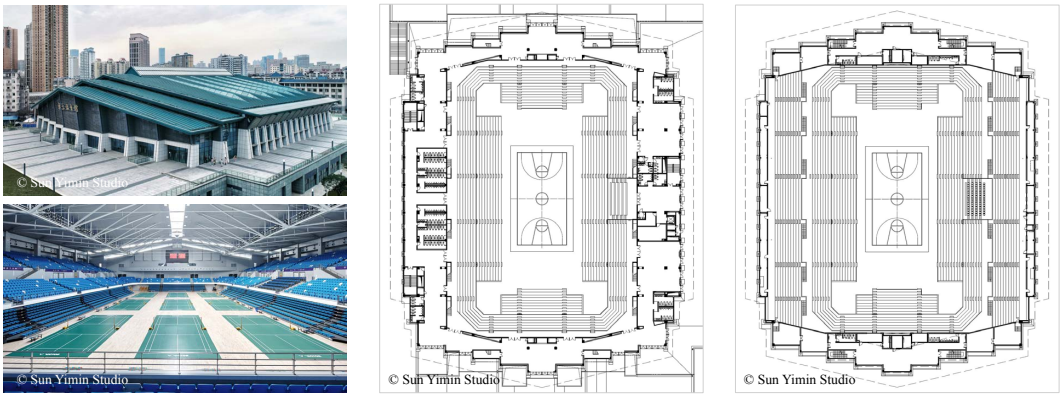


FIG. 5.2 Wuhan University gymnasium and the floor plans (revised from Yang et al., 2018)

Note: The exterior and interior views (left); the first-floor plan (middle); the second-floor plan (right). Image Source: Sun Yimin Studio.

This case study is based on a real project – Wuhan University gymnasium (see FIG.5.2). The project is designed by an interdisciplinary team, including the author, in Sun Yimin Studio of Architectural Design & Research Institute of South China University of Technology.

TOC

The site of the project is located in Wuhan, China. According to Chinese building codes – Code for thermal design of civil building (GB 50176-2016) and Standard for daylighting design of buildings (GB 50033-2013), the project is in the Hot Summer and Cold Winter climate zone and the IV daylighting climate zone. More specifically, it is located on the Wuhan University campus in a historic district filled with Chinese traditional buildings.

The project contains a competition hall with grandstands. The size of the court is 40m × 70m, which meets the requirements of many dry sports activities (e.g., basketball, badminton, gymnastics). This case study manipulates the geometries of the roofs, skylights, external shadings, roof structures, and grandstands of the hall, in order to meet architectural, daylighting, structural, energy, and thermal performances.

5.3 Phase-I involving one initial concept

Phase-I (i.e., Optimization Problem Initial-Formulation) of the Subtype-I method (i.e., non-dynamic method) involves one initial concept.

In Case Study I, the initial concept is a stair-like roof concept (Section 5.3.1). Based on this concept, a geometric parametric model and multi-disciplinary simulation or calculation models are created and integrated (Section 5.3.2), thus formulating an initial MOO model as the main output of this phase (Section 5.3.3).

5.3.1 Initial concept generation

The initial concept is a stair-like roof concept proposed at the outset of the conceptual design (see FIG.5.3). This concept is actually a typical top daylighting concept known as Monitor Skylights (CIBSE, 1999; Beltran, 2005; Harntaweegonsa and Beltran, 2007; Lechner, 2014; Al-Obaidi and Rahman, 2016; Mavridou and Doulos, 2019). It is proposed, given that the stair-like roof shape can facilitate the use of daylight and blend into the surroundings filled with Chinese traditional buildings. For simplicity, the skylights placed on the roof ridge in the real project are not considered here.

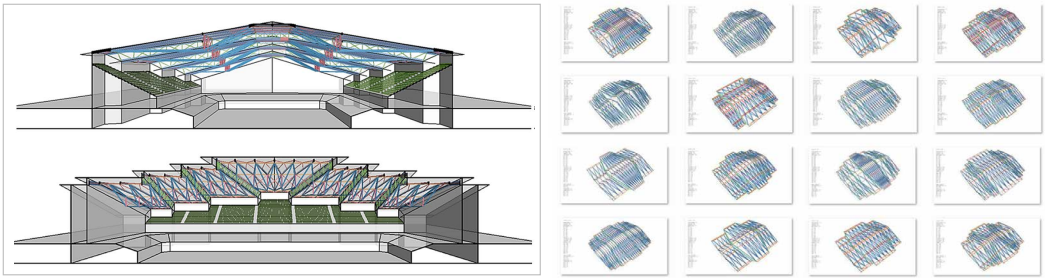


FIG. 5.3 The initial concept in Case Study I (revised from Yang et al., 2018)

Note: The stair-like roof concept (left), possible geometric variations generated based on the concept (right).

5.3.2 Multi-objective optimization model initial formulation

5.3.2.1 Geometric parametrization

A geometric parametric model is created, based on the initial stair-like roof concept illustrated in FIG.5.4. The concept implies a vast number of possible building geometries. The complexity level of the geometries is similar to the real project, given that the geometries include four parts - the geometries of the grandstands, roof envelope, external shadings, and roof structure.

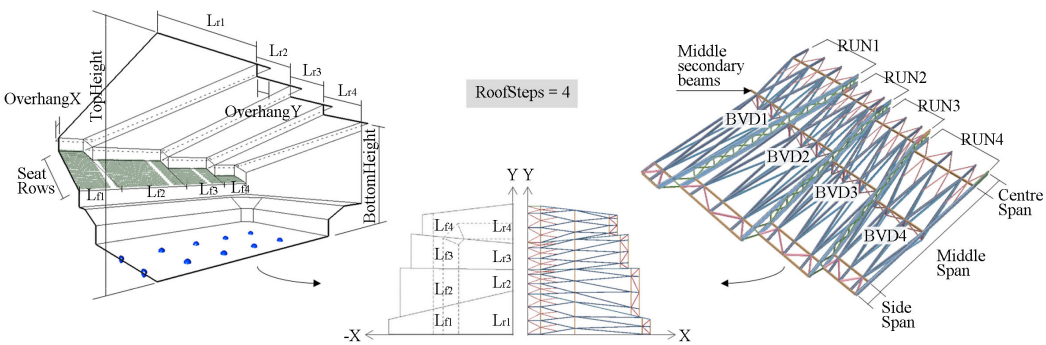


FIG. 5.4 The initial parametric model in Case Study I (revised from Yang et al., 2018)

Note: The high-level variable is "Roofsteps" which defines different roof steps in a quarter of the hall. The grandstands, roof envelope, external shadings of the initial parametric model (left); the roof structure of the initial parametric model (right); the above building parts of the initial parametric model viewed on the XY plane (middle).

TOC

TABLE 5.1 The initial design variables in Case Study I (Yang et al., 2018)

Building Parts	Variable Family	Variable Full Name	Variable Short Name	Data Type	Lower Bound	Upper Bound	Inter-vals	Bench-mark
Grand-stand	Seat Row Number	Seat row number (of the upper tier)	SeatRows	int.	15 (19)	20 (24)	1	11
Building envelope	Roof Step Number	Number of roof steps	RoofSteps	int.	2	5	1	2
	Roof Height	Height of the highest ridge (m)	TopHeight	float	25.00 (27.00)	30.00 (32.00)	0.01	26.00
		Height of the lowest ridge (m)	BottomHeight	float	15.00 (17.00)	20.00 (22.00)	0.01	24.00
	Ridge Division	Portion of the ridge of sub-roof 1	R1	float	0.20	0.90	0.01	0.9
		Portion of the ridge of sub-roof 2	R2	float	0.20	0.90	0.01	0.2
		Portion of the ridge of sub-roof 3	R3	float	0.20	0.90	0.01	-
		Portion of the ridge of sub-roof 4	R4	float	0.20	0.90	0.01	-
		Portion of the ridge of sub-roof 5	R5	float	0.20	0.90	0.01	-
	Front Row Division	Portion of the front row under sub-roof 1	F1	float	0.20	0.90	0.01	0.9
		Portion of the front row under sub-roof 2	F2	float	0.20	0.90	0.01	0.2
		Portion of the front row under sub-roof 3	F3	float	0.20	0.90	0.01	-
		Portion of the front row under sub-roof 4	F4	float	0.20	0.90	0.01	-
		Portion of the front row under sub-roof 5	F5	float	0.20	0.90	0.01	-
External shading	Shading Dimension	Overhang depth in X axis (m)	OverhangX	float	0.10	3.00	0.01	3.80
		Overhang depth in Y axis (m)	OverhangY	float	0.10	3.00	0.01	2.20

>>>

TABLE 5.1 The initial design variables in Case Study I (Yang et al., 2018)

Building Parts	Variable Family	Variable Full Name	Variable Short Name	Data Type	Lower Bound	Upper Bound	Inter-vals	Bench-mark
Roof structure	Span Partition	Centre Span (m)	CentreSpan	float	0.50	5.00	0.01	4.20
		Middle Span Partition (fraction)	MiddleSpan	float	0.10	0.90	0.01	0.50
		Side Span (m)	SideSpan	float	0.50	5.00	0.01	4.20
	Beam Vertical Distance	Beam vertical distance for sub-roof 1 (m)	BVD1	float	2.00	7.00 (6.00)	0.01	4.60
		Beam vertical distance for sub-roof 2 (m)	BVD2	float	2.00	7.00 (6.00)	0.01	2.00
		Beam vertical distance for sub-roof 3 (m)	BVD3	float	2.00	7.00 (6.00)	0.01	-
		Beam vertical distance for sub-roof 4 (m)	BVD4	float	2.00	7.00 (6.00)	0.01	-
		Beam vertical distance for sub-roof 5 (m)	BVD5	float	2.00	7.00 (6.00)	0.01	-
	Repeated Unit Number	Repeated unit number for sub-roof 1	RUN1	int.	1	5	1	5
		Repeated unit number for sub-roof 2	RUN2	int.	1	5	1	1
		Repeated unit number for sub-roof 3	RUN3	int.	1	5	1	-
		Repeated unit number for sub-roof 4	RUN4	int.	1	5	1	-
		Repeated unit number for sub-roof 5	RUN5	int.	1	5	1	-

Note: The variable marked in dark green represents the high-level variable. The variables marked in light green represent the low-level variables that can be reduced or added depending on the value of the high-level variable, while those marked in medium green represent the low-level variables that persist regardless of the value of the high-level variable. The values in brackets are the new variable bounds of the updated MOO model that is described in Section 5.4.1.1.

The geometries are initially parameterized by the design variables shown in TABLE 5.1. Given that the competition hall is symmetric along the X and Y axes, only a quarter of the hall is parametrically defined. The design variables are organized in a two-level hierarchical structure, to facilitate the geometric parameterization of the four building parts. More details are provided below.

Geometric parameterization of the grandstands

The geometric parameterization of the grandstands specifically refers to the creation of parametric schemata which define the geometries of two upper tiers having stair-like boundaries. The parameterization is carried out using a parametric grandstand design tool developed based on shape grammar (Sun et al., 2013).

For defining the upper tier, the variable SeatRows is used (see FIG.5.4, left), apart from other input geometries (e.g., the front-row profile, back-row profile, focus point, aisle axis) and input parameters (e.g., the C-value, row distance, aisle width, exit width, seat width, seat number in a row). This variable determines how far the far-back edge (row) of the upper tier is away from the front edge (row). To achieve the stair-like boundary, the back edges (i.e., those parallel to the Y axis at the back of the upper tier) are located evenly horizontally and vertically; and their number is equal to the number of sub-roof envelopes. That is, the horizontal and vertical distances between the far-back edge of the upper tier and the front edge are equally divided by the number of sub-roof envelopes; then, the results are used as the horizontal and vertical spacing between adjacent back edges. Moreover, the lengths of the back edges are determined by the geometries of sub-roof envelopes (which will be explained below).

Geometric parameterization of the roof envelope

The geometric parameterization of the roof envelope specifically refers to the creation of parametric schemata which define the geometry of a stair-like roof envelope consisting of varying sub-roof envelopes. The parameterization involves defining the division and elevation of the roof envelope.

For defining the division of the roof envelope, the variable "*RoofSteps*" and the variables in the "*Ridge division*" and "*Front edge division*" families are used (see FIG.5.4, left), to calculate the length of the ridge of each sub-roof envelope (denoted by L_{ri} and $L_{r1\text{-}r5}$), and the length of the upper tier front edge segment under each sub-roof envelope (denoted by L_{fi} and $L_{f1\text{-}f5}$), according to the equations (1) and (2). The results displayed on the XY axes can be used to determine the division of the roof envelope (see FIG.5.4, middle).

For defining the elevation of the roof envelope, the variables TopHeight and BottomHeight are used (see FIG.5.4, left), to calculate the equal spacing between adjacent ridges (denoted by S), according to the equation (3). The results can be used to determine the height of the ridge of each sub-roof envelope.

$$L_{ri} = \frac{L_r * R_i}{\sum_{j=1}^{n} R_j} \quad (1)$$

$$L_{fi} = \frac{L_f * F_i}{\sum_{j=1}^{n} F_j} \quad (2)$$

$$S = \frac{H_t - H_b}{n-1} \quad (3)$$

Where

L_r = half-length of the entire ridge (i.e., a constant value: 48 meters).

L_f = half-length of the entire upper tier front edge (i.e., a constant value: 44.5 meters).

R_i or R_j = the portion of the ridge of a sub-roof envelope *i* or each sub-roof envelope *j* (i.e., the variables in the "*Ridge division*" family).

F_i or F_j = the portion of the upper tier front edge segment under a sub-roof envelope *i* or each sub-roof envelope *j* (i.e., the variables in the "*Front edge division*" family).

H_t = the height of the ridge of the highest sub-roof envelope (i.e., the variable TopHeight).

H_b = the height of the ridge of the lowest sub-roof envelope (i.e., the variable BottomHeight).

n = number of sub-roof envelopes or roof steps (i.e., the variable "*RoofSteps*").

The bounds and intervals of the above variables are tuned carefully, to avoid too small sub-roof envelopes, and too small gaps between sub-roof envelopes, while maintaining rich variability of the roof.

A two-level hierarchical variable structure is used to facilitate the exploration of the sub-roof envelopes that vary in number and shape. The high-level variable is the

variable "*RoofSteps*" marked in dark green in TABLE 5.1. The low-level variables include those in the "*Ridge division*" and "*Front edge division*" families marked in light green and those in the "*Roof height*" family marked in medium green in TABLE 5.1. When the value of the "*RoofSteps*" changes, a different set of variables in the "*Ridge division*" and "*Front edge division*" families are selected automatically to define the geometries of the sub-roof envelopes. In this way, the geometric complexity of the roof can change considerably in a convenient way.

The geometries of the sub-roof envelopes and the geometries of the upper tiers use each other as reference. On one hand, the horizontal length of each sub-roof envelope is calculated by using the horizontal location of the associated upper tier back edge; and the height of the lowest edge of each sub-roof envelope is calculated by using the vertical location of the associated upper tier back edge. On the other hand, the length of each upper tier back edge is determined by the length of the lowest edge of the associated sub-roof envelope.

Geometric parameterization of the roof structure

The geometric parameterization of the roof structure specifically refers to the creation of parametric schemata which define the geometry of a large-span steel structure for the stair-like roof envelope. The parameterization involves defining the span of the structure, the numbers of repeated structural units, and the vertical distances between beams.

For defining the span of the structure, the variables CentreSpan and SideSpan are used (see FIG.5.4, right), to divide a half span into three parts along the X axis; either of the variables is identical for each sub-roof structure. The variable MiddleBeam is used (see FIG.5.4, middle), to determine the position of the middle beam of the lowest sub-roof structure, within the middle part of a half span; a lower value of this variable means that the middle beam of the lowest sub-roof structure is closer to the building center, and the middle beams of the other sub-roof structures align with the beam of the lowest sub-roof structure on the Y axis.

For defining the numbers of repeated structural units and the vertical distances between beams, the variables in the "*Repeated unit number*" and "*Beam vertical distance*" families are used (see FIG.5.4, right). In the either family, the variables are independent of each other; each of the variables relates to the structure of a different sub-roof.

The bounds and intervals of the above variables are tuned based on some structural rules of thumb, to avoid unfeasible design solutions, while maintaining rich variability of the structure. For instance, the variable bounds in the "*Repeated unit number*" family can ensure that the top-width of each structural unit is in between 3 to 9 meters, avoiding a too narrow or too wide unit.

A two-level hierarchical variable structure is used to facilitate the exploration of the sub-roof structures that vary together with the sub-roof envelopes. The high-level variable is still the variable "*RoofSteps*" marked in dark green in TABLE 5.1. The low-level variables include those in the "*Repeated unit number*" and "*Beam vertical distance*" families marked in light green and those in the "*Span partition*" family marked in medium green in TABLE 5.1. When the value of the "*RoofSteps*" changes, a different set of variables in the "*Repeated unit number*" and "*Beam vertical distance*" families are selected automatically to define the geometries of the sub-roof structures. In this way, the geometric complexity of the roof structure can change considerably in a convenient way.

The geometries of the sub-roof structures use the geometries of the sub-roof envelopes as a reference. Specifically, all the structural members are created by using the boundaries of the sub-roof envelopes.

Geometric parameterization of the external shadings

The geometric parameterization of the external shadings specifically refers to the creation of parametric schemata which define the geometries of roof overhangs.

For defining the roof overhangs, the variables OverhangX and OverhangY are used (see FIG.5.4, left). They represent the amounts that the roof hangs over the top of the siding in the X and Y directions (i.e., east-west and north-south directions) respectively. Either of the variables is identical for each sub-roof envelope.

The geometries of the roof overhangs use the geometries of the sub-roof envelopes as a reference. Specifically, the roof overhangs are created by extending the sub-roof envelopes outwards.

5.3.2.2 Simulation integration

Multi-disciplinary simulation or calculation models are integrated with the geometric parametric model. The initial concept is meant to meet a set of multi-disciplinary performance requirements, including architectural, daylight, thermal, energy, and structural requirements. The completeness level of the requirements is similar to the real project. The requirements are initially represented by the performance measures shown in TABLE 5.2. There can be multiple performance measures for the same kind of performance requirement. They can be considered as objectives or constraints. More details are provided below.

TABLE 5.2 The initial performance measures in Case Study I (Yang et al., 2018)

Disciplines		Performance Measures	Objectives	Constraints (to be calculated)	Constraints (set in models)
Architecture		C-value	-	-	60 mm
		Number of seats in the upper tier	-	> 3600	-
		Minimum space check (SC)	-	> 15m	-
Climate	Daylight	Modified Useful Daylight Illuminance (UDI_{mod})	↑	-	-
		Modified Uniformity Ratio (UR_{mod})	↑	-	-
	Thermal	Operative temperature	-	-	See TABLE 5.3
	Energy	Energy Use Intensity (EUI)	↓	-	-
Structure		Mass per square meter	↓	-	-
		Maximum utilization check (UC)	-	< 0.9 (failed members < 2% of the total)	-
		Maximum displacement check (DC)	-	< 0.3 m	-

Integration of architectural calculation

To obtain architectural performance feedback, a seat calculation model and a space calculation model are created. The inputs (i.e., geometries and parameters) and outputs (i.e., performance measures) of these models, and the software tools used for creating these models are specified below.

First, the input geometries of the calculation models include the parameterized geometries of the upper tier grandstands and roof structure (described in Section 5.3.2.1). These parametrically changeable geometries determine the layouts of the seats on the upper tier grandstands and the locations of the bottom beams of the roof structure, and hence affect the seat and space calculations.

Second, the input parameters of the calculation models only include parameters (not variables) used to define the above geometries, such as the row distance, aisle width, exit width, seat width, seat number in a row. which are set according to related building codes or rules of thumb. Moreover, it should be noted that a C-value can be considered as a performance measure for the quality of a sightline (DCMS, 2008). In this case, the C-value of 60 mm is set as a simulation input, differing from other performance measures that are often treated as simulation outputs. This means that the C-value is fixed, and that the quality of sightlines is always guaranteed regardless of settings of other parameters.

Third, the output performance measures of the calculation models include Seat Number and Clear Height. All these measures are initially treated as optimization constraints.

Fulfilling seating capacity and clear space requirements is important for indoor sports halls to ensure their basic functions. The measure Seat Number is defined as the total number of seats on the upper tier grandstands and can be used to check whether the grandstands can accommodate a desired number of spectators or not. In this case, it is constrained to be above 3600 in optimization (i.e., Cons_Seat Number > 3600). The measure Clear Height is defined as the minimum height of the space above the court and under the bottom beams and can be used to check whether the indoor space is high enough to host a desired set of sports activities or not. In this case, it is constrained to be above 15m in optimization (i.e., Cons_Clear Height > 15m).

Finally, the software tools used for creating the calculation models are Grasshopper's native components. These components are used to develop the parametric grandstand design tool (Sun et al., 2013), to facilitate the geometric parameterization and hence the seat and space calculations.

Integration of climatic simulation

To obtain climatic performance feedback, a daylight simulation model and an energy simulation model are created and coupled. The inputs (i.e., geometries and parameters) and outputs (i.e., performance measures) of these models, and the software tools used for creating these models are specified below.

First, the input geometries of the simulation models include the parameterized geometries of the upper tier grandstands, roof envelope, and external shadings (described in Section 5.3.2.1), and the non-parameterized geometries of the lower tier grandstand and partitions. The gaps between adjacent sub-roof envelopes are the main locations through which daylight and solar heat gain are received. The external shadings over the gaps can reflect direct sunlight off.

Second, the input parameters of the simulation models include those common in the daylight and energy simulation models, those specific for each of the two models, and those related to coupling the two models.

The parameters common in the daylight and energy simulation models are set as follows. The same weather file of Wuhan derived from Chinese Standard Weather Data is used; the weather file contains hourly outdoor environmental data. The same geometry to mesh conversion setting which can save simulation time is used. The same occupancy schedule which can consider the peak and off-peak use of the building is used (see FIG.5.5).

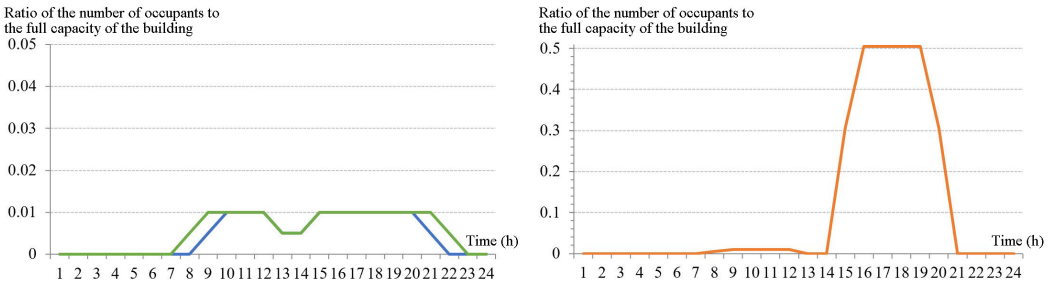


FIG. 5.5 The occupancy schedule (Yang et al., 2018)

Note: Off-peak use without spectators from Monday to Friday (blue) and on Saturday (green); peak use with spectators on Sunday (orange).

TOC

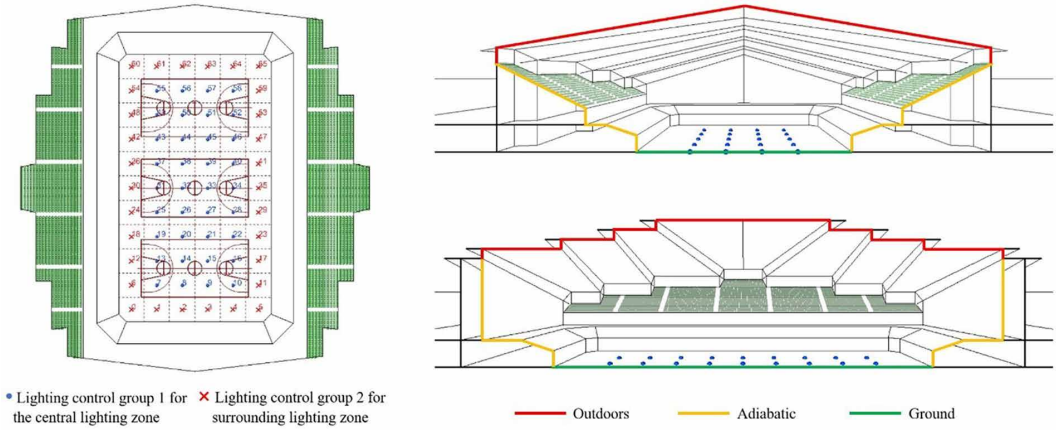


FIG. 5.6 The lighting control zones and boundary conditions (Yang et al., 2018)

Note: Two lighting zones of the court for daylight simulation (left); boundary condition for energy simulation (right).

The parameters specific for the daylight simulation model are set as follows. Two lighting control zones of the court are specified (see FIG.5.6, left). An analysis grid with 6-meter spacing is used as the basis for dividing these zones. Each cell of the grid contains an analysis point at the center; thus, 66 analysis points are used in total for the calculation of indoor illuminances. The two zones respectively cover the central and surrounding area of the court and may accommodate different sorts of activities; thus, different lighting control types and lighting power densities are set for these zones (see TABLE 5.3). Assumptions concerning optical material properties are set according to related building codes or rules of thumb (see TABLE 5.3).

The parameters specific for the energy simulation model are set as follows. Boundary conditions of the competition hall are specified (see FIG.5.6, right). Instead of modeling a full HVAC system, an ideal loads air system is used. This ideal system is assumed being able to mix air, and then add or remove heat and moisture at 100% efficiency (Big ladder, 2021). A generic heating system efficiency of 0.85 and a cooling system COP of 3 are set to scale the heating and cooling loads. Assumptions concerning thermal material properties are set according to related building codes or rules of thumb (see TABLE 5.3). Moreover, it should be noted that temperatures are actually simplified performance measures for ensuring thermal comfort. Here, setpoint and setback temperatures are set as simulation inputs, differing from other performance measures that are often treated as simulation outputs. This means that desired temperature ranges are guaranteed; in other words, thermal comfort as a constraint is always satisfied, regardless of settings of other input parameters.

TABLE 5.3 Modeling assumptions for the daylight and energy models (Yang et al., 2018)

Daylight and energy model parameter	Value
Wall reflectance	0.55
Floor reflectance	0.30
Roof reflectance	0.75
Window Transmittance	0.40
Lighting control type (lighting control group 1)	Always on during occupied hours, automatic dimming, 300-lux target
Lighting control type (lighting control group 2)	Always on during occupied hours, automatic dimming, 200-lux target
Lighting power density (lighting control group 1)	15.00 W/m^2
Lighting power density (lighting control group 2)	9.00 W/m^2
Wall U-value	0.72 W/m^2 K
Ground floor U-value	3.70 W/m^2 K
Roof U-value	0.34 W/m^2 K
Window U-value	2.60 W/m^2 K
Window SHGC	0.37
Window VT	0.62
Cooling thermostat setpoint temperature	27 °C
Cooling thermostat setback temperature	30 °C
Heating thermostat setpoint temperature	17 °C
Heating thermostat setback temperature	14 °C
Occupancy density	0.92 person/m^2
Equipment power density	2 W/m^2
Ventilation rate	15 m^3/h person
Infiltration rate	4.5 m^3/h m^2

Note: The values are from the following standards or codes: Standard for daylighting design of buildings (GB 50033-2013); Standard for lighting design of buildings (GB 50034-2013); Design code for sports building (JGJ 31-2003); Design standard for energy efficiency of public buildings (GB50189-2015); Code for thermal design of civil building (GB 50176-2016); Graduations and test methods of air permeability, watertightness, wind load resistance performance for building external windows and doors (GB/T 7106-2008).

The parameters related to coupling the daylight and energy simulation models are set as follows. This coupling is achieved by inputting lighting schedules derived from daylight simulation to energy simulation. A lighting schedule refers to a list of lighting power scalars (denoted by L) used to control a continuous dimming lighting system. The scalars are determined by the settings of a ballast loss factor and lighting setpoints, and the minimum indoor illuminance in a lighting zone, according to the equation (4). In this case, the scalars derived for either of the lighting zones, are a list of values ranging from 0.2 to 1.

$$L = \begin{cases} BLF, & if\ E_{min} \geq LS \\ BLF + (1 - BLF) * \left(1 - \frac{E_{min}}{LS}\right), & if\ E_{min} < LS \end{cases} \quad (4)$$

Where

L = lighting power scalars.

BLF = ballast loss factor (i.e., the percentage of peak energy used by a dimming system when fully dimmed down, e.g., 20%).

LS = lighting setpoints (i.e., illuminance targets for different lighting control groups, e.g., 300 lux and 200 lux respectively for the lighting control group 1 and 2, as shown in FIG.5.6, left).

E_{min} = the minimum indoor illuminance in a lighting zone.

Third, the output performance measures of the simulation models include a modified Useful Daylight Illuminance (UDI_{mod}), a modified Uniformity Ratio of Illumination (UR_{mod}), and Energy Use Intensity (EUI). All these measures are initially treated as optimization goals.

Improving daylight availability is important for indoor sports halls to reduce lighting energy use, meanwhile the amount of daylight introduced requires a proper control to avoid overheating or glare. The UDI_{mod} is a performance measure useful for measuring daylight availability and avoiding overheating or glare risks. It is actually a modified version of the original Useful Daylight Illuminance (UDI). The original UDI represents the annual occurrence of "*useful*" daylight illuminances that fall within the range of 100-2000 lux, that is, the percentage of occupied hours that an analysis point receives the "*useful*" daylight illuminances (Nabil and Mardaljevic, 2006). It is calculated based on indoor illuminances at all time steps and at one analysis point. It is difficult to understand the daylight availability condition of a larger space (e.g., the 40m × 70m court in question) when more analysis points are not considered. Thus, the original UDI is modified in order to allow multiple analysis points. The modified measure UDI_{mod}, also known as spatial UDI, is defined as the percentage of floor area (i.e., the percentage of analysis points) that receives the "*useful*" daylight illuminances for at least a specified percentage of occupied hours. It can reflect the daylight availability condition of the entire large space using a single value, thus facilitating its use in optimization. The UDI_{mod} is to be maximized (i.e., Max_UDI_{mod}).

Moreover, it should be noted that the percentage of occupied hours for which the "*useful*" daylight illuminances are received is subject to change in this case. Initially, this percentage is set to 60%, and the UDI_{mod} is more precisely denoted by UDI_{mod-60}.

Improving daylight uniformity (i.e., illumination uniformity) is important for indoor sports halls, as a uniform daylight condition can help athletes and audiences to perceive fast-moving balls. The UR_{mod} is a performance measure useful for measuring the average of daylight uniformity. It is actually a modified version of the original Uniformity Ratio (UR). The original UR represents the ratio between minimum illuminance and average illuminance. It is calculated based on indoor illuminances at all analysis points and at one time step. It is difficult to understand the general daylight uniformity condition in a longer time span (e.g., the time span of a year) when more time steps are not considered. Thus, the original UR is modified in order to allow multiple time steps. The modified measure UR_{mod} is defined as the mean of UR values at all time steps of a year when daylight is available. It can be used as an objective in optimization to improve average UR values. The UR_{mod} is to be maximized (i.e., Max_UR_{mod}).

Reducing operational energy use is important for indoor sports halls to achieve sustainable daily operation. The EUI is a performance measure useful for measuring operational energy use. It represents annual energy use (for lighting, heating, and cooling energy etc.) per square meter of floor area. It can aid in benchmarking the energy efficiency of buildings. The EUI is to be minimized (i.e., Min_EUI).

Finally, the software tools used for creating the simulation models are Grasshopper's plug-ins called Ladybug and Honeybee which adopt Daysim and EnergyPlus simulation engines. These two engines are combined given the following facts. On one hand, EnergyPlus shows a significant limitation in calculating indoor illuminances, that is, it tends to overestimate the amount of daylight in indoor environments (Ramos and Ghisi, 2010). On the other hand, Daysim can model automated lighting control systems and provide hourly lighting schedules of different lighting zones to calculate final energy use in EnergyPlus (Futrell et al., 2015; Didoné and Pereira, 2011).

Integration of structural simulation

To obtain structural performance feedback, a structural simulation model is created. The inputs (i.e., geometries and parameters) and outputs (i.e., performance measures) of this model, and the software tool used for creating this model are specified below.

First, the input geometries of the simulation model include the parameterized geometries of the roof structure (described in Section 5.3.2.1). The large-span steel roof structure is the main load-bearing structure. The central and largest-span sub-roof structure (see FIG.5.7) approximately spans over 91.6 meters between the farthest supports (i.e., the farthest locations the upper tiers can reach). According to the structural functions and practical engineering considerations, the structural elements are grouped into different types, as shown by the color coding. Steel beams in a diamond pattern in two layers form a one-way span steel frame, which is the main feature of the structural system. Steel cables are applied in the lateral direction providing lateral stability at multiple locations. A space truss is used at the gap area where two sub-roof envelops at different elevations interface with each other.

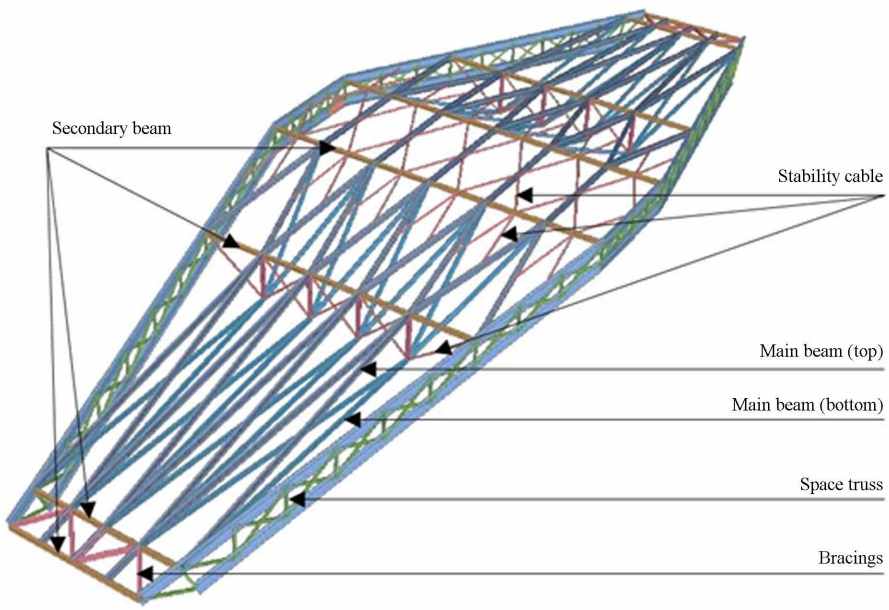


FIG. 5.7 Typical structure for one sub-roof (Yang et al., 2018)

TOC

Second, the input parameters of the simulation model include those in relation to sections, buckling, and loads.

The parameters related to sections are set as follows. Each group of structural elements is only assigned with one identical section, in order to simplify the connection design and reduce the number of different joints. A list of standard steel section profiles (including HE beam and rod section profiles) is set for each group. S355 steel is set as the steel grade for all the section profiles.

The parameter related to buckling is set as follows. Unimportant secondary structural members which connect the main beams are not modelled directly in the buckling calculation, in order to increase the computational speed and simplify the model. To account for this, a reduction factor on the buckling length is introduced, and set for the lateral torsional buckling and the minor axis bending buckling calculation of the main beams.

The parameters related to loads are set as follows. The loads are defined based on Eurocode. The load combinations are set to involve the most typical loads, such as the structure's self-weight, super imposed dead load, wind load, and snow load.

Third, the output performance measures of the simulation model include Steel Use Intensity (SUI), Utility, and Displacement. The SUI is initially treated as an optimization goal, the Utility and Displacement are initially treated as optimization constraints.

Reducing the use of roof structure steel is important for indoor sports halls to reduce embodied energy usage. The SUI is a performance measure useful for measuring roof structure steel use. It represents the mass of a steel roof structure divided by total floor area. It can help benchmark the steel use efficiency of buildings. The SUI is to be minimized (i.e., Min_SUI).

Fulfilling Ultimate Limit State (ULS) and Service Limit State (SLS) requirements is important for indoor sports halls to ensure the safety of the large-span structure. The measure Utility and Displacement can be used respectively to check whether ULS and SLS requirements are fulfilled or not. In this case, the maximum utility allowed in ULS check or Unity Check is 0.9 (i.e., Min_UC ≤ 0.9); and the maximum displacement allowed in SLS check or Displacement Check is 0.3m (i.e., Min_DC ≤ 0.3m). That is, a structural member is constrained to have a utility value less than or equal to 0.9 and a displacement value less than or equal to 0.3m. Moreover, it should be noted that a tolerance value has been introduced in the unity check in this case, in order to reduce the total number of unfeasible solutions and prevent missing many

potential designs. In some cases, once a structural member fails the unity check, the structural design is considered as an unfeasible solution. But, from a practical point of view, the occurrence of a small number of structural members which slightly go beyond the utility constraint is actually allowed, because the violation of the constraint can be easily solved afterwards by strengthening it locally. Thus, in this case, a tolerance value of 2% is applied in the unity check, meaning that 2% of the total structural members are allowed to slightly go beyond the utility constraint.

Finally, the software tool used for creating the simulation model is Grasshopper's plug-in called Karamba 3D which adopts finite element analysis (FEA) simulation methods. This tool contains a local optimization module which can automatically select the optimum section profile from among all provided ones according to EN1993; and it contains a code checking module based on EN1993 for the unity check.

5.3.3 Output of Phase-I

The main output of Phase-I is an initial MOO model which includes an initial set of performance objectives, constraints, and design variables. This model is to be used in the next phase.

5.4 Phase-II adopting a linear re-formulation process

Phase-II (i.e., Optimization Problem Re-Formulation) of the Subtype-I method (i.e., non-dynamic method) adopts a linear process.

In Case Study I, the design context is to highlight reducing existing design possibilities; thus, the linear re-formulation process adopted is a one-time re-formulation process that focuses on removing existing variables (i.e., refining an existing concept convergently). For this re-formulation, designers have to discover the answers to the following questions:

- Which performance measures are more meaningful for final objectives or constraints?

- How many “steps” of the roof are more promising to lead to good quantitative and qualitative performances?

- How, and to what extent, do the geometries of the grandstands, roof envelope, external shadings, and roof structure affect quantitative and qualitative performances?

- How to achieve a proper MOO model that has the potential to lead to better Pareto solutions?

Normally, designers answer these questions highly relying on their personal past experiences. However, as performance objectives, constraints, and design variables increase, the complexity of these questions increases rapidly. In this circumstance, it is difficult for designers to obtain the right answers to these questions by purely relying on their past experiences.

Given the above fact, it is necessary to carry out quantitative data analysis from multiple angles in this phase, in order to extract useful information and knowledge, and to support the linear re-formulation process. As exemplified in FIG.3.4, three types of actions are conducted in this phase (Section 5.4.1) for achieving a final MOO model (Section 5.4.2).

5.4.1 One-time re-formulation

5.4.1.1 Data generation

Two groups of 500 data sets are generated for analysis. Each group is derived based on an automation process consisting of a MOO model, the Uniform Latin Hypercube sampling algorithm, and the sequential execution order (see FIG.5.8). A data set contains quantitative data (i.e., numeric design values and performance values) and qualitative data (i.e., building geometries).

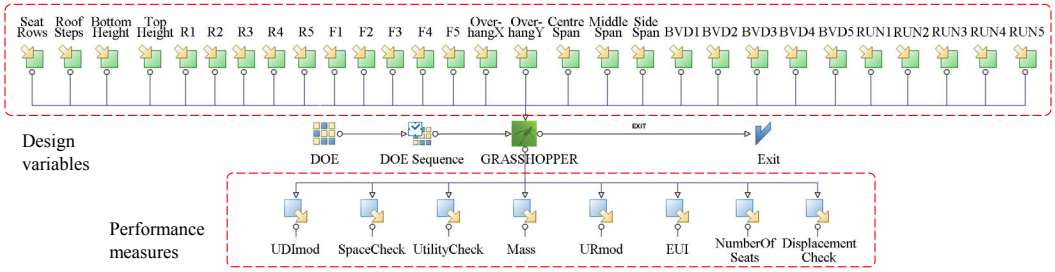


FIG. 5.8 The automation process for generating the data in Case study I (revised from Yang et al., 2018)

The initial MOO model is used to generate the first group of data sets. However, the data sets obtained are not ideal in some senses. This indicates the need to adjust the strictness of the performance measures and the bounds of the design variables, namely update the initial MOO model. The updated model is then used to generate the second group of 500 data sets. The two groups of data sets are checked, as described below.

The performance values in the first group of data sets are checked, in order to know the proportion of feasible or unfeasible solutions and the performance value distribution. First, as statistics shows, unfeasible solutions account for a major portion (87.6%), which is mainly due to the violation of architectural constraints, namely Seat Number and Clear Height constraints (see TABLE 5.4). Thus, it is necessary to adjust the bounds of relevant variables, to increase the portion of feasible solutions. In this case, the bounds of the SeatRows, TopHeight, BottomHeight, and BVD1 to BVD5 are adjusted. Second, the statistics also shows that $UDI_{mod\text{-}60}$ values are mostly distributed in the high value range, which means the measure $UDI_{mod\text{-}60}$ can be readily satisfied (see FIG.5.9, left). Thus, it is reasonable to use a stricter measure to achieve a higher daylight availability level. In this case, the stricter measure used is $UDI_{mod\text{-}65}$, which increases the percentage of occupied hours for which the "*useful*" daylight illuminances are received.

TABLE 5.4 Summary of the two groups of data sets (Yang et al., 2018)

DoE data set	Feasible solutions	Unfeasible solutions	Violation of constraints			
			Con_NOS	Con_SC	Con_UC	Con_DC
Initial DoE data set	62 (12.4%)	438 (87.6%)	261	367	57	21
Second DoE data set	230 (46.0%)	270 (54.0%)	52	169	75	30

The performance values in the second group of data sets are checked, in a similar way. As expected, unfeasible solutions are reduced significantly (see TABLE 5.4), and UDI_{mod-65} values are mostly distributed in the low value range (see FIG.5.9, right). The former can help achieve more feasible solutions by satisfying the violated architectural constraints; the latter can help to leave sufficient room for achieving higher daylight availability levels. The second 500 data sets will be used for the consequent information and knowledge extraction.

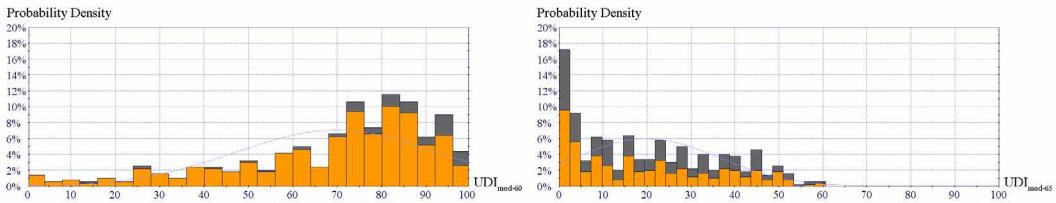


FIG. 5.9 The distribution of UDI values in the two groups of data sets (Yang et al., 2018)

Note: The distribution of UDImod-60 values in the first group of data sets (left); the distribution of UDImod-65 values in the second group of data sets (right). The orange and gray columns respectively represent the percentages of unfeasible and feasible solutions.

5.4.1.2 Information and knowledge extraction

Based on the data sets obtained, three categories of knowledge are extracted, namely knowledge about the performance measures, the high-level variable, and the low-level variables, as exemplified in FIG.3.4.

Knowledge about the performance measures

To acquire knowledge about which quantitative performance measures are more meaningful for final objectives or constraints, it is helpful to extract correlations between related measures. However, extracting such correlations accurately is challenging, especially when there are many measures to be considered, and/or, when the data sets used for the correlation extraction are changed.

To handle this challenge, a specific correlation analysis technique - Pearson Correlation (Glen, 2021) is used. The dimensions of the analyzed data sets consist of the measures EUI, SUI, UDI_{mod-65}, and UR_{mod}. As a result, correlation coefficients

between all the measures are generated and visualized in a correlation matrix chart (see FIG.5.10, top). The numerical values in the lower left of the chart represent the correlation coefficients; the images in the upper right of the chart represent the related 2D scatter plots; and the bar charts running diagonally between these areas represent the discrete probability density functions of all the measures. Based on the result, correlation information is extracted and interpreted, so as to acquire desired knowledge, as described below.

The extracted correlation information includes: (1) EUI does not correlate with the other three measures, as the absolute correlation coefficients are smaller than 0.1; (2) SUI has a weak correlation with $UDI_{mod\text{-}65}$ and UR_{mod}, as the absolute correlation coefficients are in between 0.1 and 0.3; (3) $UDI_{mod\text{-}65}$ has a medium correlation with UR_{mod}, as the absolute correlation coefficient is in between 0.3 and 0.5 (see FIG.5.10, bottom); and (4) $UDI_{mod\text{-}65}$ and UR_{mod} have the same desired changing directions – increasing their values as much as possible.

The extracted correlation information is interpreted in disciplinary contexts. Measures from different disciplines may be weakly or not correlated. In this case, the climate-related measures EUI, $UDI_{mod\text{-}65}$, and UR_{mod} are functions of variables which define the roof envelope; and the structure-related measure SUI is a function of variables which define the roof structure. Thus, these two types of measures are probably weakly or not correlated, given that they are functions of different variables. Measures from the same discipline may be notably correlated. In this case, a high $UDI_{mod\text{-}65}$ value indicates that too low and too high illuminance levels are mostly avoided in the court area; and avoiding the extreme illuminance levels can help to improve daylight uniformity and achieve a high UR_{mod} value. Thus, $UDI_{mod\text{-}65}$ may be positively correlated with UR_{mod} to a noticeable extent. Measures from the same discipline are not necessarily notably correlated. In this case, a high $UDI_{mod\text{-}65}$ value does not definitely indicate a high or low EUI value. This mainly because a EUI value can be determined by a balance between energy saved from using daylight and energy used for cooling. Specifically, when more daylight is available, lighting energy savings and cooling energy usage may increase at the same time. The lighting energy savings can be or cannot be offset by the cooling energy usage. Thus, $UDI_{mod\text{-}65}$ are not necessarily correlated with EUI to a noticeable extent.

Based on the information extraction and interpretation, the knowledge below is acquired. The quantitative measures $UDI_{mod\text{-}65}$, EUI, and SUI are considered to be more meaningful choices to form objectives (i.e., Max_$UDI_{mod\text{-}65}$, Min_EUI, Min_SUI); the quantitative measure UR_{mod} is considered to be a more meaningful choice to form a constraint (i.e., Cons_UR_{mod}). This is given that $UDI_{mod\text{-}65}$ and UR_{mod} are positively and notably correlated, and their desired changing directions are the same.

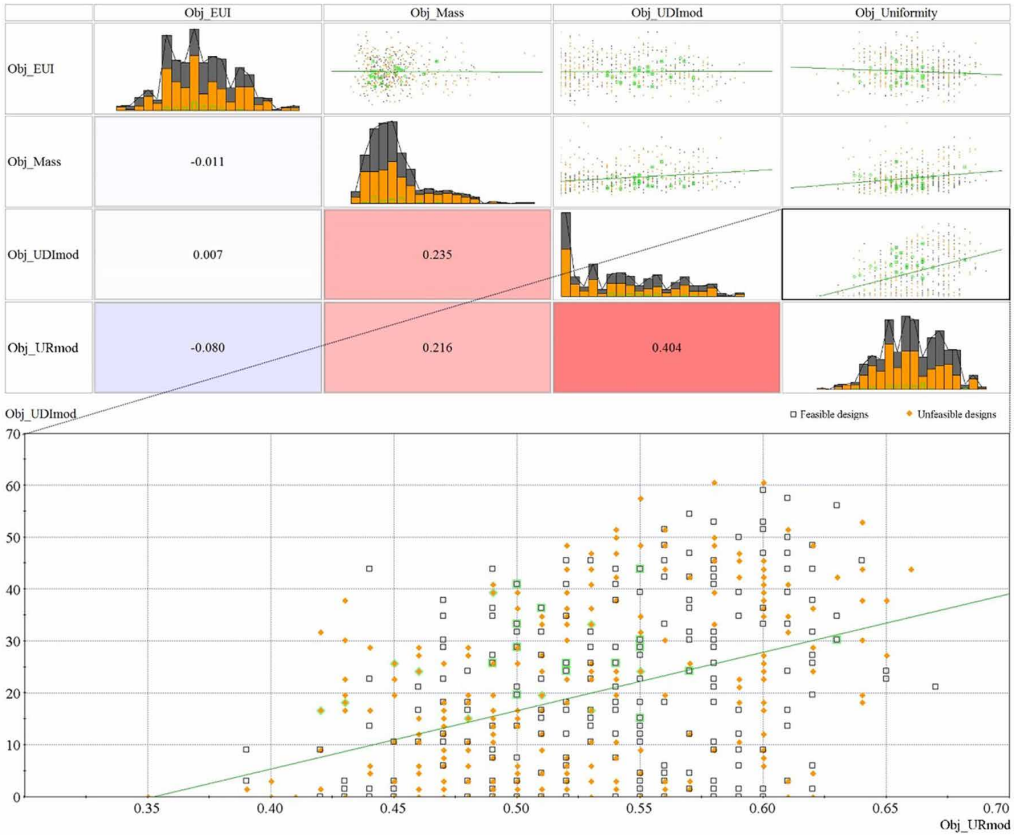


FIG. 5.10 Correlation analysis results (Yang et al., 2018)

Note: Correlation matrix chart of four candidate objective variables (left); 2D scatter plot of Obj_UDImod-65 and Obj_URmod (right).

To acquire knowledge about which qualitative performance measures are more meaningful for final objectives or constraints, human subjectivity is required (i.e., subjectively determining meaningful qualitative measures). In this case, a qualitative measure Aesthetics is considered to be a more meaningful choice to form a constraint, among many possible measures (e.g., cultural, social related measures). It is defined as the aesthetic quality of roof envelopes specifically.

Overall, in this case, the quantitative measures $UDI_{mod\text{-}65}$, EUI, and SUI are considered to be more meaningful choices to form objectives, while the other measures are considered to be more meaningful choices to form constraints. Note that, screening objectives is a decision that needs both quantitative and qualitative related knowledge. With those knowledge, designers can prioritize quantitative and qualitative measures for determining the final screening of the objectives.

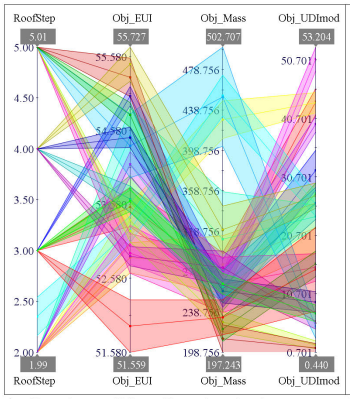
Knowledge about the high-level variable

To acquire knowledge about which high-level variable value (i.e., "*RoofSteps*" value) is more promising to lead to good EUI, SUI and UDI_{mod-65} performances, it is helpful to construct meaningful clusters of samples according to "*RoofSteps*" values and performance values. However, constructing such clusters manually is challenging, especially when there are many samples to be clustered, and/or, when there exists multiple measured characteristics for clustering.

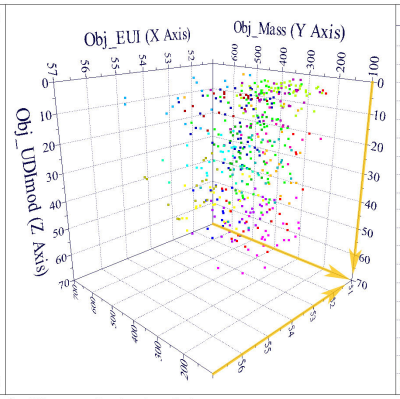
To handle the challenge, a specific cluster analysis technique - Hierarchical Clustering (Jain et al., 1999) is used. The dimensions of the analyzed data sets consist of the high-level variable "*RoofSteps*" and the measures EUI, SUI and UDI_{mod-65}. In this analysis, samples having the same or similar "*RoofSteps*" values and EUI, SUI, and UDI_{mod-65} values are expected to be grouped in a cluster. As a result, twenty clusters are constructed and visualized in a clustering parallel coordinate chart and a 3D scatter plot (see FIG.5.11, top).

The extracted clustering information includes: (1) EUI values can be rather high or low, for samples having varying roof steps, SUI values can be relatively lower, for samples having 3 and 4 roof steps, and UDI_{mod-65} values roughly decrease along with the increase of roof step numbers (see FIG.5.11, middle); (2) samples in CLUSTER_0 and CLUSTER_1 having 3 roof steps, samples in CLUSTER_5 having 2 roof steps, and samples in CLUSTER_16 having 4 roof steps can reach the chosen objectives simultaneously (see FIG.5.11, bottom); and (3) among the above four clusters of samples, those in CLUSTER_0 and CLUSTER_1 having 3 roof steps account for the major portion, and those in CLUSTER_5 having 2 roof steps are quantitatively more promising (see FIG.5.11, bottom). For extracting this information, human subjectivity is required (i.e., subjectively determining desired relative importance of the chosen objectives). This relative importance is reflected by filtering arrows (see FIG.5.11, bottom). The closer a filtering arrow reaches the end of an objective (i.e., the low-value ends of EUI and SUI, the high-value end of UDI_{mod-65}), the more important the objective is.

The extracted clustering information is interpreted in disciplinary contexts. Regarding the Roofstep-EUI relation, the number of roof steps (i.e., the width between roof steps) is not the only factor affecting energy use; there can be other important factors such as the distribution of skylights. Thus, EUI values vary significantly regardless of the number of roof steps. Regarding the Roofstep-SUI relation, when the number of roof steps is too small or large, the structural elements can become too sparse or dense, which possibly makes the structure inefficient. Thus, SUI values are low when the number of roof steps is not too small or large.



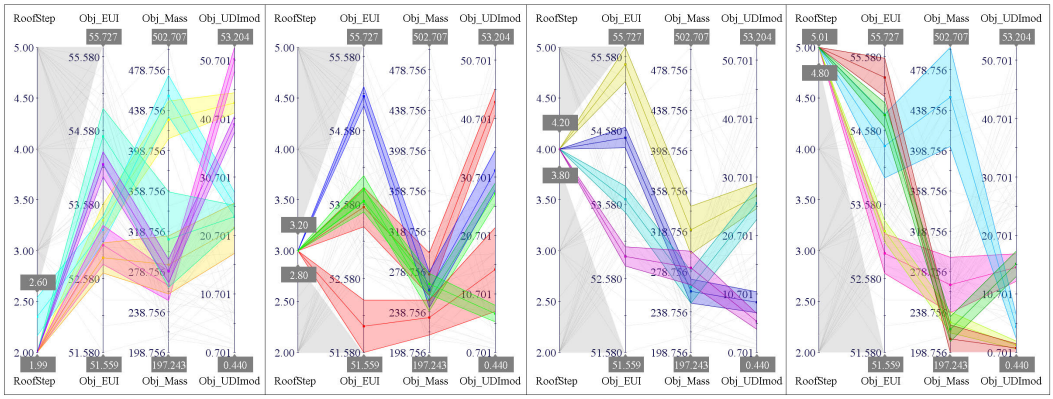
▲ Clustering parallel coordinate chart showing all clusters



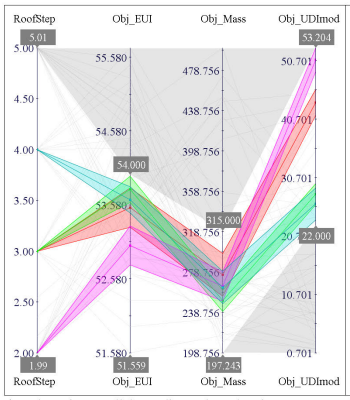
▲ 3D scatter plot showing all clusters

CATEGORY	RoofStep	Obj_EUI	Obj_Mass	Obj_UDImod
CLUSTER_0	3.00	53.55	276.16	43.63
CLUSTER_1	3.00	53.80	250.48	27.93
CLUSTER_2	3.00	55.05	260.72	31.90
CLUSTER_3	2.00	53.37	429.42	43.48
CLUSTER_4	2.35	53.24	453.25	26.06
CLUSTER_5	2.00	53.04	264.74	50.92
CLUSTER_6	2.00	52.86	285.55	21.97
CLUSTER_7	2.00	54.50	310.70	23.95
CLUSTER_8	2.00	54.12	279.36	39.80
CLUSTER_9	5.00	52.92	265.62	15.26
CLUSTER_10	5.00	53.22	228.55	1.95
CLUSTER_11	5.00	54.37	452.07	5.09
CLUSTER_12	5.00	55.30	212.37	1.44
CLUSTER_13	5.00	54.79	221.89	15.88
CLUSTER_14	4.00	54.49	259.49	9.30
CLUSTER_15	4.00	55.48	320.24	27.54
CLUSTER_16	4.00	53.65	263.73	26.23
CLUSTER_17	4.00	52.88	282.57	5.98
CLUSTER_18	3.00	51.93	233.50	14.90
CLUSTER_19	3.00	53.63	266.65	7.37

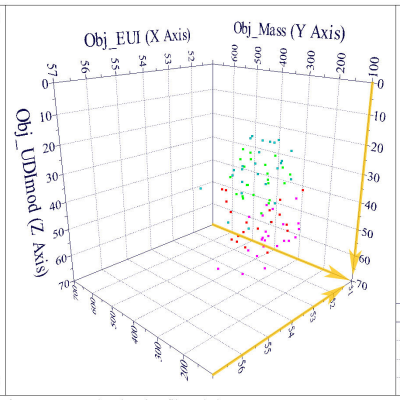
▲ Mean values of RoofStep, Obj_EUI, Obj_Mass, Obj_UDImod of all clusters



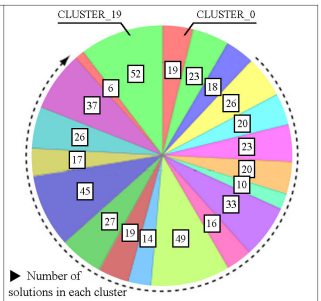
▲ Clustering parallel coordinate chart showing clusters with a similar number of roof steps; in each chart from left to right, the number of roof steps equals to (or approximately equals to) 2, 3, 4 or 5, respectively.



▲ Clustering parallel coordinate chart showing filtered clusters



▲ 3D scatter plot showing filtered clusters



▶ Number of solutions in each cluster

CATEGORY	RoofStep	Obj_EUI	Obj_Mass	Obj_UDImod
CLUSTER_0	3.00	53.55	276.16	43.63
CLUSTER_1	3.00	53.80	250.48	27.93
CLUSTER_5	2.00	53.04	264.74	50.92
CLUSTER_16	4.00	53.65	263.73	26.23

▲ Mean values of RoofStep, Obj_EUI, Obj_Mass, Obj_UDImod of filtered clusters

FIG. 5.11 Hierarchical clustering results (Yang et al., 2018)

Note: All twenty clusters created via the cluster analysis (top); the filtered clusters with a similar number of roof steps (middle); the filtered clusters with desired performance trends (bottom).

TOC

Regarding the Roofstep-$UDI_{mod\text{-}65}$ relation, when the number of roof steps is large, daylight can be more evenly spread over the indoor space, which possibly results in some more spots with insufficient illuminance. Thus, $UDI_{mod\text{-}65}$ values decrease as the increase of roof step numbers.

Based on the information extraction and interpretation, the knowledge below is acquired. The high-level variable values 2, 3, and 4 are considered more promising to lead to good EUI, SUI, and $UDI_{mod\text{-}65}$ performances. This is mainly given the fact that samples in CLUSTER_0, CLUSTER_1, CLUSTER_5, and CLUSTER_16 can reach the chosen EUI, SUI and $UDI_{mod\text{-}65}$ objectives simultaneously.

To acquire knowledge about which high-level variable value (i.e., "*RoofSteps*" value) is more promising to lead to acceptable Aesthetics performance, human subjectivity is required (i.e., subjectively evaluating the Aesthetics performance of clusters of samples). In this case, clusters of samples having 2, 3, and 4 roof steps are assumed aesthetically more promising, that is, the high-level variable values 2, 3, and 4 are considered more promising to lead to acceptable Aesthetics performance.

Overall, in this case, the high-level variable values 2, 3, and 4 are considered more promising, any of which can be chosen to determine the low-level variables in question. Note that, choosing a high-level variable value is a decision that needs both quantitative and qualitative related knowledge. If a high-level variable value promising to achieve good quantitative performances is different from that promising to achieve acceptable qualitative performances, the two types of performance measures need to be prioritized first, before determining the final choice of the high-level variable value.

Knowledge about the low-level variables

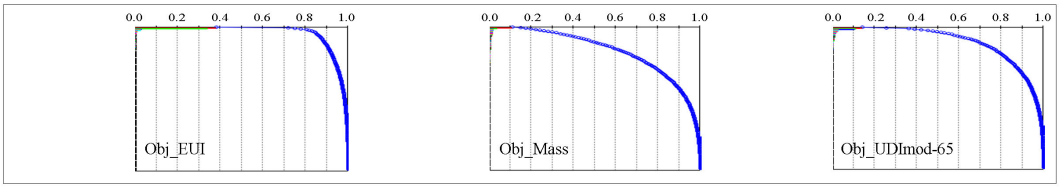
To acquire knowledge about which low-level variables are more important to the variance of EUI, SUI, and $UDI_{mod\text{-}65}$ performances, it is helpful to extract the sensitivity of these variables on the performances. However, extracting such sensitivity accurately is challenging, especially when there are many variables involved, and/or, when both main and interaction effects (Rahman, 2019) of the variables are to be considered.

To handle the challenge, a specific sensitivity analysis technique - Smoothing Spline Analysis of Variance (Gu, 2002; Ricco et al., 2013) is used. The dimensions of the analyzed data sets consist of the low-level variables (excluding BVD4, BVD5, F4, F5, R4, R5, RUN4, RUN5, as "*RoofSteps*" value equals to 3) and the measures EUI, SUI,

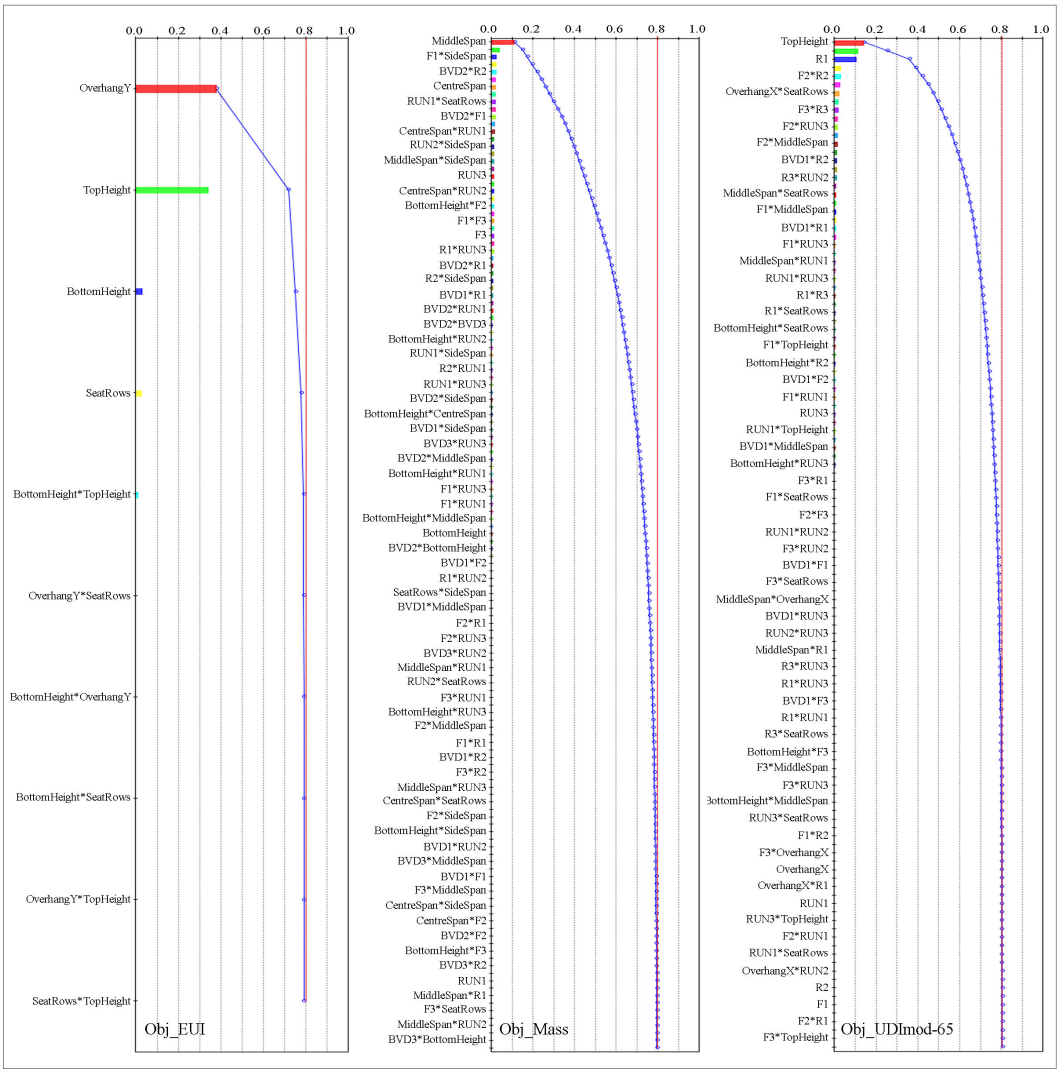
and $UDI_{mod\text{-}65}$. In this analysis, the variables and measures are also called "*factors*" and "*responses*" respectively; a main-effect factor or an interaction-effect factor is also called a "*term*". As a result, the relative importance (i.e., the percentage of contribution) of each term to the global variance of a response is extracted and visualized in an effect column chart (see FIG.5.12).

The extracted sensitivity information includes: (1) the contributions of the terms to EUI are very diverse, the most important two factors (i.e. OverhangY and TopHeight) are responsible for the major portions (i.e. 37.9% and 33.9% respectively) of the variance of EUI; (2) the contributions of the terms to SUI are less diverse, the most important factor (i.e. MiddleSpan) is responsible for a small portion (i.e.10.8%) of the variance of SUI, the remaining terms as a whole account for the major portion; (3) the contributions of the terms to $UDI_{mod\text{-}65}$ are also less diverse, the most important three factors (i.e. TopHeight, BottomHeight, and R1) are respectively responsible for a small portion (i.e., 14.2%, 11.2%, and 10.5%) of the variance of $UDI_{mod\text{-}65}$, the remaining terms as a whole account for a larger portion; and (4) interaction effects are non-negligible, especially for SUI and $UDI_{mod\text{-}65}$. For extracting this information, human subjectivity is required (i.e., subjectively determining desired percentages of the cumulative effect of unremoved terms). These percentages are reflected by red vertical lines (see FIG.5.12). In this case, 80% of the cumulative effect is used to identify unremoved terms; the factors involved in these unremoved terms are all considered as important variables, as shown in light green in TABLE 5.5.

The extracted sensitivity information is interpreted in disciplinary contexts. Since OverhangY defines the overhang depth for north-south-facing clearstories, it can significantly affect the amount of daylight coming into the interior space; and since TopHeight defines the height and hence volume of the interior space, it can significantly affect the amount of cooling loads required to maintain certain thermal conditions. Thus, these two factors can have large effects to the variance of EUI. Since MiddleSpan defines the locations of maximum vertical distances between upper and lower main beams, it can affect the load-bearing capacity of the roof structure. Thus, this factor can have a noticeable effect to the variance of SUI. Since TopHeight and BottomHeight define the vertical locations and sizes of clearstories, and R1 defines the horizontal locations of clearstories, they can affect the daylight availability levels on the court. Thus, these three factors can have a noticeable effect to the variance of $UDI_{mod\text{-}65}$.



▲ Effect column charts showing all main and interaction terms that maintain a cumulative effect of 100%
(note: all the names of the terms are hidden, as the focus of the charts is to show the complete cumulative effects)



▲ Effect column charts showing the main and interaction terms that maintain a cumulative effect of about 80%
(note: some names of the terms are hidden, due to the limited space)

FIG. 5.12 Sensitivity analysis results (Yang et al., 2018)

Note: All terms that maintain a cumulative effect of 100% (top); the terms that maintain a cumulative effect of 80% (bottom).

TABLE 5.5 Important design variables for each response and their main effects (Yang et al., 2018)

	Seat Rows	Bottom Height	Top Height	R1	R2	R3	F1	F2	F3	Over-hang X	Over-hang Y	Centre Span	Middle Span	Side Span	BVD1	BVD2	BVD3	RUN1	RUN2	RUN3
Obj_EUI	0.026	0.032	0.339	0.026	0.002	0.000	0.012	0.000	0.000	0.004	0.379	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000
Obj_Mass	0.013	0.002	0.002	0.019	0.000	0.000	0.003	0.000	0.000	0.003	0.002	0.020	0.108	0.021	0.012	0.012	0.000	0.000	0.039	0.012
Obj_UDImod-65	0.009	0.112	0.142	0.105	0.000	0.005	0.000	0.000	0.017	0.000	0.000	0.000	0.000	0.001	0.004	0.006	0.001	0.000	0.000	0.003

Based on the information extraction and interpretation, the knowledge below is acquired. All the low-level variables are considered important to the variance of EUI, SUI, and $UDI_{mod\text{-}65}$ performances. This is mainly given the fact that any of these variables (i.e., factors) is involved at least in one unremoved term for one of the responses.

To acquire knowledge about which low-level variables are more important to the variance of Aesthetics performance, human subjectivity is required (i.e., subjectively determining the importance of the low-level variables on Aesthetics performance). In this case, all the low-level variables are assumed to be able to affect Aesthetics performance, that is, being important to the variance of Aesthetics performance.

Overall, in this case, all the low-level variables in question are considered important to the variance of the quantitative and qualitative performances. Note that, screening low-level variables is a decision that needs both quantitative and qualitative related knowledge. If low-level variables important to quantitative performances are different from those important to qualitative performances, the two kinds of performance measures need to be prioritized first, before determining the final screening of the low-level variables.

5.4.1.3 Multi-objective optimization model re-formulation

The initial MOO model is re-formulated based on the acquired knowledge. The specific re-formulation actions include quantitative objective reduction, qualitative constraint addition, high-level variable reduction, and low-level variable reduction, as suggested by the acquired knowledge.

For different purposes of study, different MOO models can be created. In this study, MOO model 0-5 are created by selecting different sets of performance objectives, constraints, and design variables (see TABLE 5.6 and TABLE 5.7), as described below.

TABLE 5.6 Re-formulated optimization problems and the execution (revised from Yang et al., 2018)

	MOO models	Optimization problem (re-)formulation				Execution		
		Number of objectives	Number of (Type I) design variables	Number of (Type II & III) design variables	Initial generation	Actual evaluated designs	Total time	Time per design
Traditional method	MOO model 0	4	1 (RoofSteps = 2,3,4,5)	16-28	ULH	462	37h:43m	4.90m
Proposed method	MOO model 1	3	0 (RoofSteps = 3)	20	High-performing cluster [a]	463	42h:02m	5.45m
Factor 1	MOO model 2	2	0 (RoofSteps = 3)	20	High-performing cluster [a]	453	40h:51m	5.41m
Factor 2	MOO model 3	3	0 (RoofSteps = 2)	16	High-performing cluster [b]	455	32h:01m	4.22m
	MOO model 4	3	0 (RoofSteps = 4)	24	High-performing cluster [c]	449	51h:41m	6.91m
Factor 3	MOO model 5	3	0 (RoofSteps = 3)	18	High-performing cluster [d]	463	42h:52m	5.56m
Factor 4	MOO model 1	3	0 (RoofSteps = 3)	20	ULH	465	44h:09m	5.70m

High-performing cluster [a]: CLUSTER_0 and CLUSTER_1 having "*RoofSteps*" value 3.

High-performing cluster [b]: CLUSTER_5 having "*RoofSteps*" value 2.

High-performing cluster [c]: CLUSTER_16 having "*RoofSteps*" value 4.

High-performing cluster [d]: CLUSTER_0 and CLUSTER_1 having "*RoofSteps*" value 3 (R3 and RUN3 are treated as constants in these clusters).

TABLE 5.7 Lists of re-formulated objectives and design variables (Yang et al., 2018)

	Traditional method	Proposed method	Factor1	Factor2	Factor2	Factor3	Factor4
MOO models	MOO model 0	MOO model 1	MOO model 2	MOO model 3	MOO model 4	MOO model 5	MOO model 1
Objectives	Obj_EUI	Obj_EUI	Obj_EUI	Obj_EUI	Obj_EUI	Obj_EUI	Obj_EUI
	Obj_Mass	Obj_Mass	Obj_Mass	Obj_Mass	Obj_Mass	Obj_Mass	Obj_Mass
	$Obj\_UDI_{mod\text{-}65}$	$Obj\_UDI_{mod\text{-}65}$	-	$Obj\_UDI_{mod\text{-}65}$	$Obj\_UDI_{mod\text{-}65}$	$Obj\_UDI_{mod\text{-}65}$	$Obj\_UDI_{mod\text{-}65}$
	Obj_UR_{mod}	-	-	-	-	-	-
Design variables (Type I)	RoofSteps	-	-	-	-	-	-
Design variables (Type II & III)	SeatRows	SeatRows	SeatRows	SeatRows	SeatRows	SeatRows	SeatRows
	BottomHeight	BottomHeight	BottomHeight	BottomHeight	BottomHeight	BottomHeight	BottomHeight
	TopHeight	TopHeight	TopHeight	TopHeight	TopHeight	TopHeight	TopHeight
	R1	R1	R1	R1	R1	R1	R1
	R2	R2	R2	R2	R2	R2	R2
	R3	R3	R3	-	R3	-	R3
	R4	-	-	-	R4	-	-
	R5	-	-	-	-	-	-
	F1	F1	F1	F1	F1	F1	F1
	F2	F2	F2	F2	F2	F2	F2
	F3	F3	F3	-	F3	F3	F3
	F4	-	-	-	F4	-	-
	F5	-	-	-	-	-	-
	OverhangX	OverhangX	OverhangX	OverhangX	OverhangX	OverhangX	OverhangX
	OverhangY	OverhangY	OverhangY	OverhangY	OverhangY	OverhangY	OverhangY
	CentreSpan	CentreSpan	CentreSpan	CentreSpan	CentreSpan	CentreSpan	CentreSpan
	MiddleSpan	MiddleSpan	MiddleSpan	MiddleSpan	MiddleSpan	MiddleSpan	MiddleSpan
	SideSpan	SideSpan	SideSpan	SideSpan	SideSpan	SideSpan	SideSpan
	BVD1	BVD1	BVD1	BVD1	BVD1	BVD1	BVD1
	BVD2	BVD2	BVD2	BVD2	BVD2	BVD2	BVD2
	BVD3	BVD3	BVD3	-	BVD3	BVD3	BVD3
	BVD4	-	-	-	BVD4	-	-
	BVD5	-	-	-	-	-	-
	RUN1	RUN1	RUN1	RUN1	RUN1	RUN1	RUN1
	RUN2	RUN2	RUN2	RUN2	RUN2	RUN2	RUN2
	RUN3	RUN3	RUN3	-	RUN3	-	RUN3
	RUN4	-	-	-	RUN4	-	-
	RUN5	-	-	-	-	-	-

The first purpose is to study whether the Subtype-I method (i.e., non-dynamic method) is better than the traditional method (defined in FIG.3.1). For this, MOO model 0 is created based on the traditional method. This means that all the initial performance objectives, constraints, and design variables are kept unchanged to create the model. Moreover, MOO model 1 is created based on the acquired knowledge. First, the desired quantitative objective number is set to 3, as the quantitative objective Max_UR_{mod} is transformed into a quantitative constraint Cons_UR_{mod} whose lower bound is 0.58 (i.e., the third quartile of the UR_{mod} values); second, the desired "*RoofSteps*" value is set to 3, thus the high-level variable "*RoofSteps*" is transformed into a constant value, and irrelevant low-level variables BVD4, BVD5, F4, F5, R4, R5, RUN4, and RUN5 are removed; third, the desired percentages of cumulative effects of important terms are set to 80%, thus none of the remaining low-level variables are transformed into a constant value; and finally, a qualitative constraint Cons_Aesthetics is added.

The second purpose is to study how the choice of the objectives may affect the optimization results. For this, MOO model 2 is created. First, the desired quantitative objective number is set to 2, as the quantitative objective Max_$UDI_{mod\text{-}65}$ is transformed into a quantitative constraint Cons_$UDI_{mod\text{-}65}$ whose lower bound is 32.55% (i.e., the third quartile of the $UDI_{mod\text{-}65}$ values); then, the other aspects are the same as those for MOO model 1.

The third purpose is to study how the choice of the high-level variable values (i.e., "*RoofSteps*" values) may affect the optimization results. For this, MOO model 3 and 4 are created. First, the desired "*RoofSteps*" values are set to 2 and 4, thus the high-level variable "*RoofSteps*" in each of the models is transformed into a constant value, and irrelevant low-level variables are removed; then, the other aspects are the same as those for MOO model 1.

The fourth purpose is to study how the choice of the low-level variables may affect the optimization results. For this, MOO model 5 is created. First, the desired percentages of cumulative effects of important terms (for SUI and $UDI_{mod\text{-}65}$) are set to 60%, thus the low-level variables R3 and RUN3 are transformed into constant values (i.e., the medians of the R3 and RUN3 values); the other aspects are the same as those for MOO model 1.

The fifth purpose is to study how the use of a purely random initial population may affect the optimization results. For this, MOO model 1 is used.

5.4.2 Output of Phase-II

The main output of Phase-II includes a final MOO model which includes a final set of performance objectives, constraints, and design variables (i.e., MOO model 1). Moreover, it can also include other MOO models for comparison purposes. In this case, all of them are to be used in the next phase.

5.5 Phase-III utilizing a directed initial population

Phase-III (i.e., Optimization Problem Solving) of the Subtype-I method (i.e., non-dynamic method) utilizes a directed initial population.

In Case Study I, the directed initial population consists of samples selected from high-performing clusters. Such clusters can be found from the parallel coordinate chart shown in FIG.5.11. In order to study relevant hypotheses, different initial population (i.e., directed and purely random initial populations) are used in combination with different MOO models. They are used to set up multiple MOO runs which are then executed (Section 5.5.1). MOO results are compared (Section 5.5.2) in order to extract knowledge about the hypotheses as the main output of this phase (Section 5.5.3).

5.5.1 Multi-objective optimization setup and execution

5.5.1.1 Setup for studying hypothesis one

It is hypothesized that adopting the Subtype-I method (i.e., non-dynamic method) can help achieve a quantitatively and qualitatively better Pareto front, compared with adopting the traditional method (defined in FIG.3.1). For studying this hypothesis, two groups of MOO runs are set up (i.e., MOO run A and B), as described below.

For studying the effects of adopting the traditional method, MOO run A is set up. It is based on: MOO model 0 (where the chosen "*RoofSteps*" values are 2, 3, 4, and 5), and a purely random initial population (selected from the entire design space).

For studying the effects of adopting the Subtype-I method (i.e., non-dynamic method), MOO run B is set up. It is based on: MOO model 1 (where the chosen "*RoofSteps*" value is 3), and a directed initial population (selected from CLUSTER_0 and CLUSTER_1 having "*RoofSteps*" value 3).

5.5.1.2 Setup for studying hypothesis two

It is hypothesized that factors including the choice of objectives, high-level variable values, low-level variables, and initial populations may affect the behaviors of Subtype-I method (i.e., non-dynamic method). More specifically, factors of over-screening objectives, choosing different high-level variable values (i.e., "*RoofSteps*" values), over-screening low-level variables, and utilizing a purely random initial population may affect final MOO results. For studying this hypothesis, five more groups of MOO runs are set up (i.e., MOO run C to G), as described below.

For studying the impacts of over-screening objectives, MOO run C is set up. It is based on: MOO model 2 (where the chosen "*RoofSteps*" value is 3, and more objectives are removed), and a directed initial population (selected from CLUSTER_0 and CLUSTER_1 having "*RoofSteps*" value 3).

For studying the impacts of choosing different high-level variable values (i.e., "*RoofSteps*" values), MOO run D and E are set up. MOO run D is based on: MOO model 3 (where the chosen "*RoofSteps*" value is 2), and a directed initial population (selected from CLUSTER_5 having "*RoofSteps*" value 2). MOO run E is based on: MOO model 4 (where the chosen "*RoofSteps*" value is 4), and a directed initial population (selected from CLUSTER_16 having "*RoofSteps*" value 4).

For studying the impacts of over-screening low-level variables, MOO run F is set up. It is based on: MOO model 5 (where the chosen "*RoofSteps*" value is 3, and more low-level variables are removed), and a directed initial population (selected from CLUSTER_0 and CLUSTER_1 having "*RoofSteps*" value 3).

For studying the impacts of using a purely random initial population, MOO run G is set up. It is based on: MOO model 1, and a purely random initial population (selected from the entire design space).

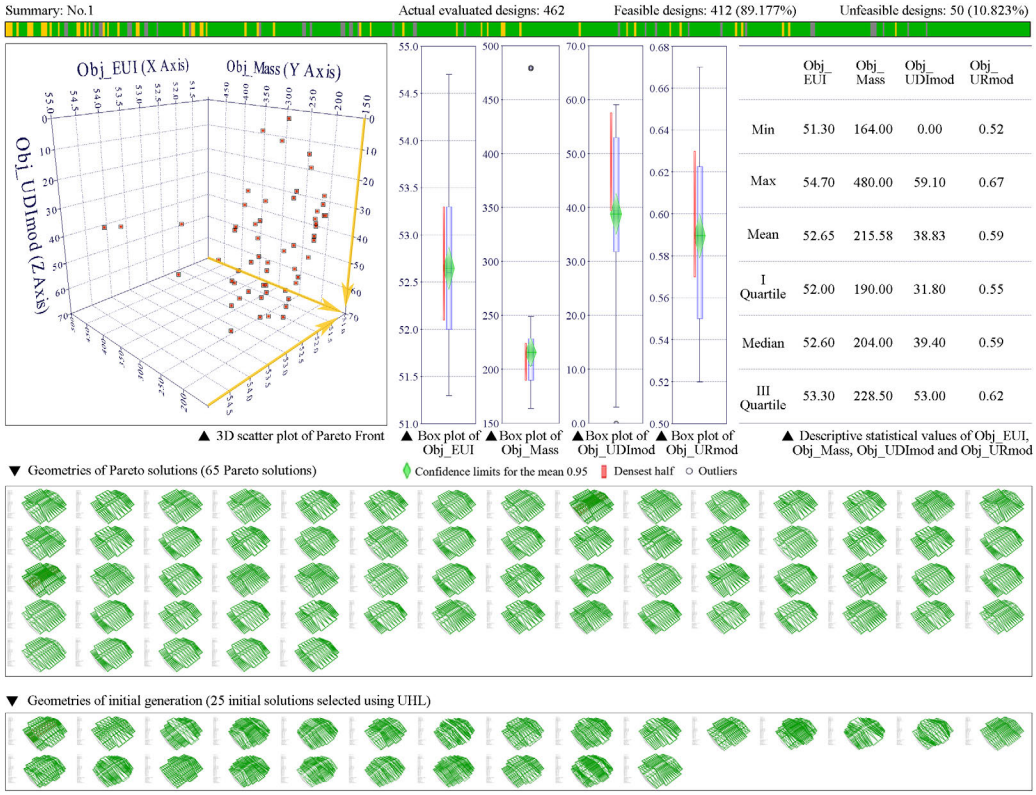
5.5.1.3 Optimization execution

There is a total of seven optimizations to execute. For all these optimizations, the same MOO algorithm and settings are used: Non-dominant Sorting Genetic Algorithm II (NSGA-II), a population size of 25, and 20 generations etc. All these optimizations are executed on a 6-Core (12-Thread) computer. The total time for executing each optimization is around 2 days; and the average time for evaluating each solution is around 5 minutes.

5.5.2 Multi-objective optimization result comparison

The MOO results include the (quantitative and qualitative) data of Pareto solutions. The data is organized in ways that facilitate the comparison of Pareto solutions (see FIG.5.13 - FIG.5.19). For instance, the performance values of Pareto solutions are plotted in the same 3D space and summarized using box-whisker plots and tables; the geometries of Pareto solutions are presented next to each other; and the numbers of Pareto solutions, unfeasible solutions, and broken solutions that violates certain constraints are presented in Appendix V. Moreover, the optimization result comparison is summarized in TABLE 5.8.

Some issues are important for comparing the MOO results. First, it is important to know the concept of a hypervolume indicator (Zitzler and Thiele, 1998; Zitzler et al., 2007). This indicator can be used to compare different Pareto fronts. A higher hypervolume value represents a better Pareto front in terms of proximity and diversity. To allow meaningful comparisons, the same reference point is used to calculate hypervolume values. Moreover, it is also important to make clear what an ideal Pareto front is in the context that highlights reducing existing design possibilities. As explained in Section 3.2.3.3, an ideal Pareto front in this context should have good proximity and diversity, good pertinence, good geometric preference compliance, and good geometric variation appropriateness. Here, good pertinence means that Pareto solutions are within a small region of interest; good geometric variation appropriateness means that Pareto solutions have a low degree of geometric variations.



	Obj_ EUI	Obj_ Mass	Obj_ UDImod	Obj_ URmod
Min	51.30	164.00	0.00	0.52
Max	54.70	480.00	59.10	0.67
Mean	52.65	215.58	38.83	0.59
I Quartile	52.00	190.00	31.80	0.55
Median	52.60	204.00	39.40	0.59
III Quartile	53.30	228.50	53.00	0.62

FIG. 5.13 Optimization results derived from MOO run A in Case Study I (Yang et al., 2018)

TOC

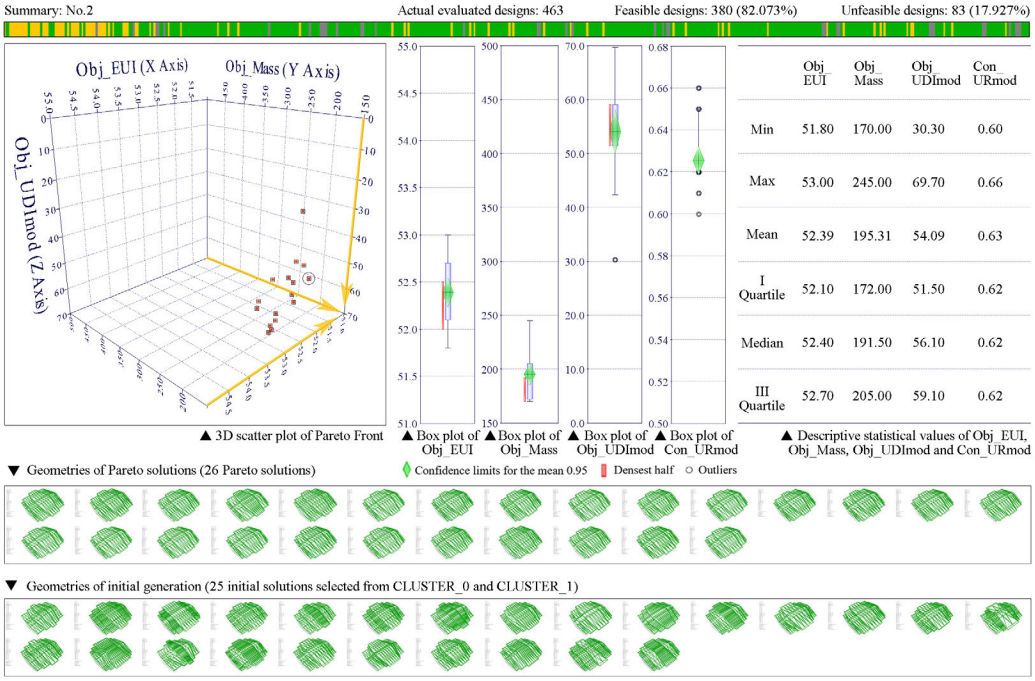


FIG. 5.14 Optimization results derived from MOO run B in Case Study I (Yang et al., 2018)

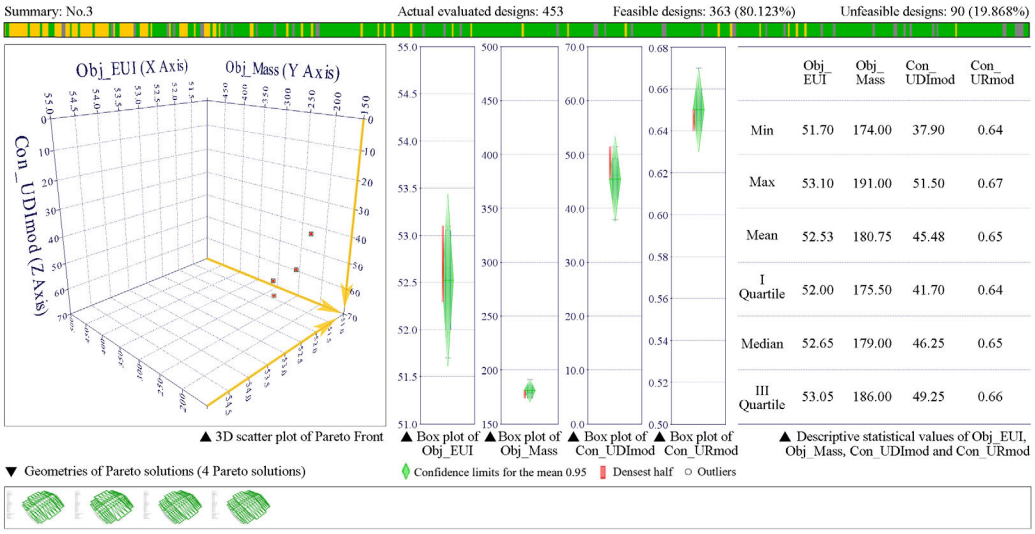


FIG. 5.15 Optimization results derived from MOO run C in Case Study I (Yang et al., 2018)

TOC

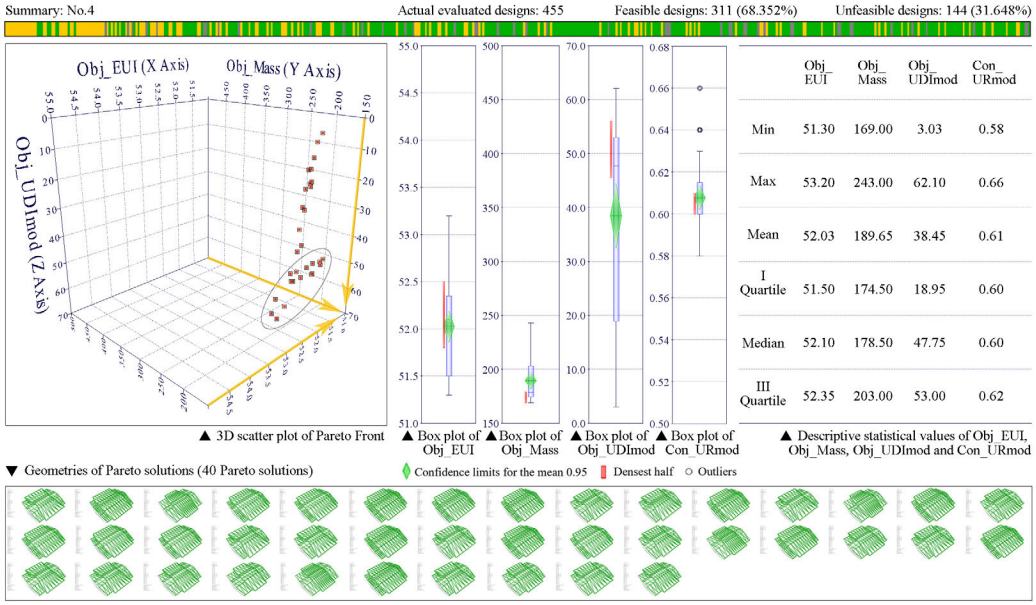


FIG. 5.16 Optimization results derived from MOO run D in Case Study I (Yang et al., 2018)

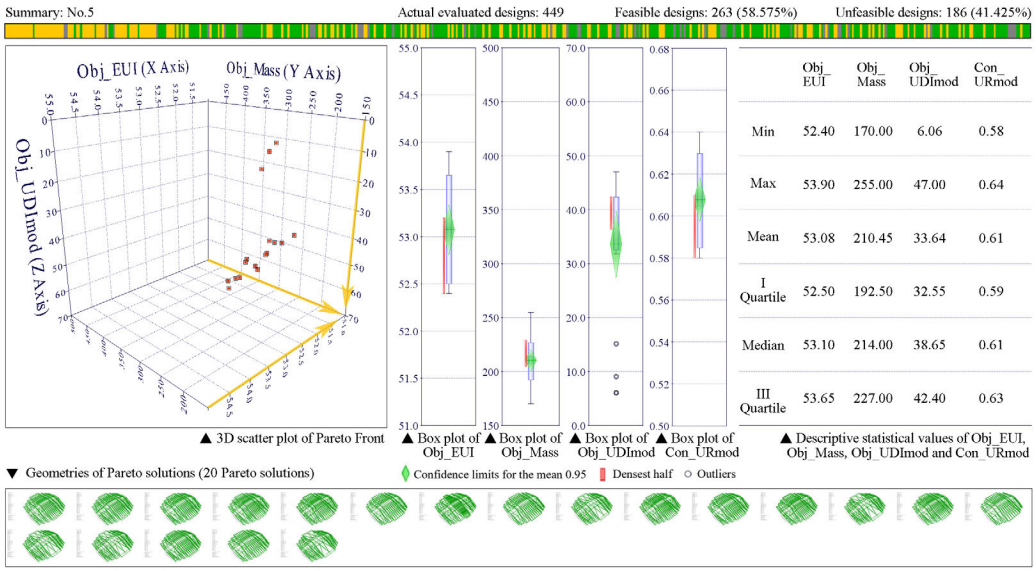


FIG. 5.17 Optimization results derived from MOO run E in Case Study I (Yang et al., 2018)

TOC

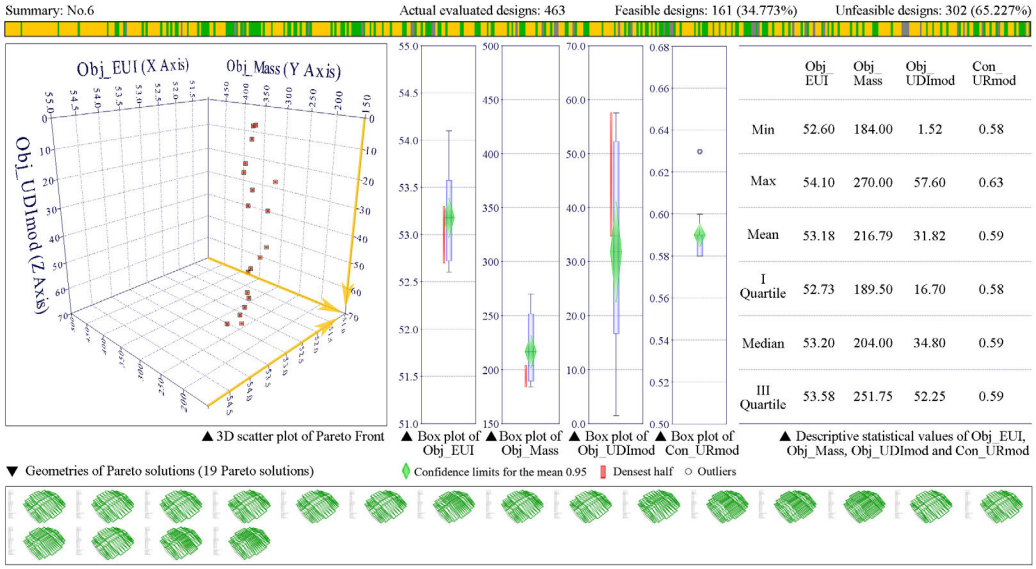


FIG. 5.18 Optimization results derived from MOO run F in Case Study I (Yang et al., 2018)

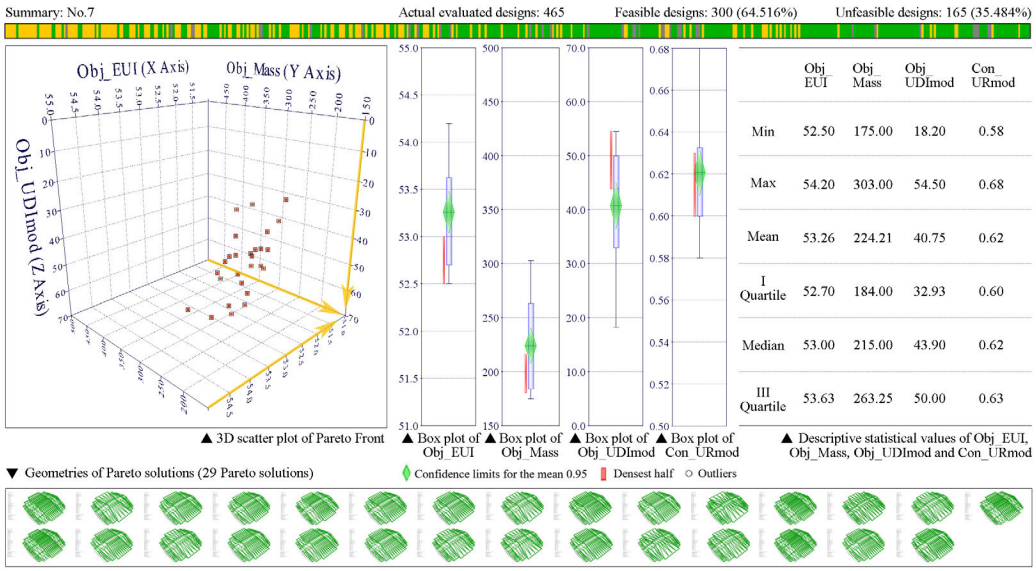


FIG. 5.19 Optimization results derived from MOO run G in Case Study I (Yang et al., 2018)

TOC

TABLE 5.8 Summary of optimization result comparison in Case Study I

Purposes of the verification	Optimization result comparison	Quantity	Pertinence	Proximity and diversity	Geometric preference compliance	Geometric variation appropriateness
Verify the benefits of adopting the Subtype-I method	Pareto solutions from MOO run B (compared with those from MOO run A)	+	+	+	x	+
Verify the impacts of over-screening objectives	Pareto solutions from MOO run C (compared with those from MOO run B)	-	+	-	x	x
Verify the impacts of choosing different high-level variable values (i.e., "RoofSteps" values)	Pareto solutions from MOO run D (compared with those from MOO run B)	-	-	+	x	-
	Pareto solutions from MOO run E (compared with those from MOO run B)	-	-	-	x	-
Verify the impacts of over-screening low-level variables	Pareto solutions from MOO run F (compared with those from MOO run B)	-	-	-	x	-
Verify the impacts of utilizing a purely random initial population	Pareto solutions from MOO run G (compared with those from MOO run B)	-	-	-	x	-

Note: "+" represents better results or positive impacts; "-" represents worse results or negative impacts; "x" represents similar results or no significant impacts.

5.5.2.1 Result comparison for verifying hypothesis one

To verify the benefits of adopting the Subtype-I method (i.e., non-dynamic method), the Pareto solutions from MOO run B (see FIG.5.14) are compared with the Pareto solutions from MOO run A (see FIG.5.13).

The Pareto solutions from MOO run B (1) have a quantity more similar to the size of the initial population; (2) are within a smaller region of interest; (3) lead to a higher hypervolume value 67999.83; (4) are similarly compliant with the geometric preference; and (5) have a lower degree of geometric variations. These facts confirm that adopting the Subtype-I method (i.e., non-dynamic method) can help achieve a better Pareto front in terms of the quantity, pertinence, proximity and diversity, and geometric variation appropriateness, compared with adopting the traditional method.

It is also worthwhile to compare an individual solution from MOO run B with that from the real project. Specifically, a random Pareto solution derived from executing MOO run B (circled in FIG.5.14) and a benchmark solution most like the real project (listed

in the last column of TABLE 5.1) are chosen for the comparison. The data of these two solutions are shown in FIG.5.20. It is observed that the EUI, Mass, UDI_{mod-65}, and UR_{mod} performances of the former solution are better than those of the latter solution, especially the UDI_{mod-65} performance. This implies the need of adding additional skylights for the latter solution to increase daylight availability, as shown in the real project. It is also observed that the geometry of the former solution is significantly different from that of the latter solution. This indicates the possibility of obtaining more creative designs by providing more freedom to design exploration.

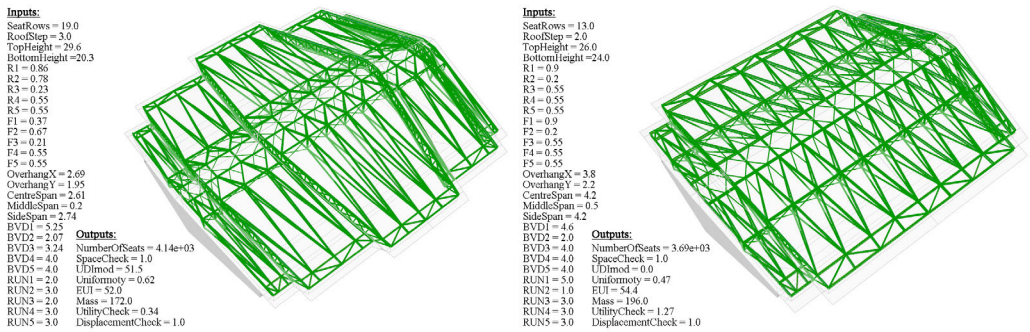


FIG. 5.20 Comparison of an optimal solution and a real project solution (Yang et al., 2018)

Note: A random Pareto solution derived from executing MOO run B (left); a benchmark solution most like the real project (right).

5.5.2.2 Result comparison for verifying hypothesis two

To verify the impacts of over-screening objectives, the Pareto solutions from MOO run C (see FIG.5.15) are compared with the Pareto solutions from MOO run B (see FIG.5.14). The Pareto solutions from MOO run C (1) have a quantity less similar to the size of the initial population; (2) are within a smaller region of interest; (3) lead to a lower hypervolume value 51050.11; (4) are similarly compliant with the geometric preference; and (5) have a similar degree of geometric variations. These facts confirm that over-screening objectives can have negative impacts on the quantity, proximity and diversity of the Pareto front, but have positive impacts on the pertinence.

To verify the impacts of choosing different high-level variable values (i.e., "*RoofSteps*" values), the Pareto solutions from MOO run D (see FIG.5.16) and the Pareto solutions from MOO run E (see FIG.5.17) are respectively compared with the Pareto solutions from MOO run B (see FIG.5.14).

First, the Pareto solutions from MOO run D (1) have a quantity less similar to the size of the initial population; (2) are within a larger region of interest; (3) lead to a higher hypervolume value 71355.90; (4) are similarly compliant with the geometric preference; and (5) have a higher degree of geometric variations. These facts confirm that choosing "*RoofSteps*" value 2 can have negative impacts on the quantity, pertinence, and geometric variation appropriateness of the Pareto front, but have positive impacts on the proximity and diversity.

Second, the Pareto solutions from MOO run E (1) have a quantity less similar to the size of the initial population; (2) are within a larger region of interest; (3) lead to a lower hypervolume value 36706.78; (4) are similarly compliant with the geometric preference; and (5) have a higher degree of geometric variations. These facts confirm that choosing "*RoofSteps*" value 4 can have negative impacts on the quantity, pertinence, proximity and diversity, and geometric variation appropriateness of the Pareto front.

To verify the impacts of over-screening low-level variables, the Pareto solutions from MOO run F (see FIG.5.18) are compared with the Pareto solutions from MOO run B (see FIG.5.14). The Pareto solutions from MOO run F (1) have a quantity less similar to the size of the initial population; (2) are within a larger region of interest; (3) lead to a lower hypervolume value 34768.40; (4) are similarly compliant with the geometric preference; and (5) have a higher degree of geometric variations. These facts confirm that over-screening low-level variables can have negative impacts on the quantity, pertinence, proximity and diversity, and geometric variation appropriateness of the Pareto front.

To verify the impacts of utilizing a purely random initial population, the Pareto solutions from MOO run G (see FIG.5.19) are compared with the Pareto solutions from MOO run B (see FIG.5.14). The Pareto solutions from MOO run G (1) have a quantity less similar to the size of the initial population; (2) are within a larger region of interest; (3) lead to a lower hypervolume value 39746.10; (4) are similarly compliant with the geometric preference; and (5) have a higher degree of geometric variations. These facts confirm that utilizing a purely random initial population can have negative impacts on the quantity, pertinence, proximity and diversity, and geometric variation appropriateness of the Pareto front.

5.5.3 Output of Phase-III

The main output of Phase-III includes not only final Pareto solutions, but also various knowledge derived by comparing these solutions. In this case, the knowledge obtained in this phase is about the benefits of adopting the Subtype-I method (i.e., non-dynamic method) and the factors affecting the behaviors of the method.

5.6 Discussion

In this case study, the benefits of adopting the Subtype-I method (i.e., non-dynamic method) are derived from the incorporation of the linear knowledge-supported re-formulation process (Section 5.4). During the knowledge extraction process, convergent knowledge is obtained (i.e., the knowledge for helping a design process to converge on an acceptable design solution). With the help of this knowledge, unimportant design variables are removed to refine a concept convergently, thus facilitating the achievement of a more reliable optimization model and more reliable optimal solutions.

Human designers are influential during the linear re-formulation process. Their influence actually has two sides. On one side, they have advantages to support qualitative thinking; while on the other side, they may cause negative impacts on final optimal results when using the Subtype-I method (i.e., non-dynamic method) in different ways. For instance, when designers decide to choose a high-level variable value (i.e., "*RoofSteps*" value) within those that are considered quantitatively promising, both positive and negative impacts on the final Pareto front can occur; when designers decide to reduce the number of low-level variables extensively, negative impacts on the final Pareto front can occur; when designers decide to utilize a purely random initial population (rather than a directed initial population), the starting point for optimization is relatively low, which can also lead to negative impacts on the final Pareto front.

5.7 Conclusion

This chapter concludes by summarizing the main research results (Section 5.6.1); identifying possible extensions of the Subtype-I method (i.e., non-dynamic method) (Section 5.6.2); and providing concluding remarks (Section 5.6.3).

5.7.1 Main research results

The main research results of this chapter include the following:

- The overall process of the Subtype-I method (i.e., non-dynamic method) has been demonstrated in Case Study I. This subtype method includes a one-time Optimization Problem Re-Formulation (Re-OPF) process that focuses on removing existing variables (i.e., refining an existing concept convergently).

- The benefits of adopting the Subtype-I method (i.e., non-dynamic method) have been verified. As confirmed by the optimization result comparison, adopting this subtype method can help achieve a better Pareto front in terms of the quantity, pertinence, proximity and diversity, and geometric variation appropriateness, compared with adopting the traditional method.

- The factors affecting the behaviors of the Subtype-I method (i.e., non-dynamic method) have been verified. As confirmed by the optimization result comparison, factors of over-screening objectives, choosing different high-level variable values (i.e., “RoofSteps” values), over-screening low-level variables, and utilizing a purely random initial population can have varying impacts on the optimization results, including positive and negative impacts, as summarized in TABLE 5.8. Most of the impacts are negative and need to be avoided; but it is also worth noting the positive impacts, namely the impact of over-screening objectives on the pertinence of the Pareto front, and the impact of choosing different high-level variable values (i.e., “RoofSteps” values) on the proximity and diversity of the Pareto front.

5.7.2 Possible extensions

The Subtype-I method (i.e., non-dynamic method) can be further extended at least in the following two ways:

- First, this subtype method can be extended by including more initial concepts in the Optimization Problem Initial-Formulation (Initial-OPF) phase. In Case Study I, the method has only included one initial concept defined by using a hierarchical variable structure. This can fit the context that highlights reducing existing design possibilities. However, in the context that highlights sparking new design possibilities, it is meaningful to include multiple initial concepts. These concepts can be also defined by using a hierarchical variable structure. More specifically, a high-level variable "Concept" can be used to label different concepts, namely, different sets of low-level variables used to define the geometries of different concepts.

- Second, this subtype method can be extended by allowing the addition of new variables in the Optimization Problem Re-Formulation (Re-OPF) phase. In Case Study I, the method has simply incorporated a one-time re-formulation process that focuses on removing existing variables. This can fit the context that highlights reducing existing design possibilities. However, in the context that highlights sparking new design possibilities, it is meaningful to incorporate a multiple-time re-formulation process that focuses on adding new variables. The addition of new variables reflects the addition of new concepts or ideas.

5.7.3 Concluding remarks

In conclusion, the Subtype-I method (i.e., non-dynamic method) is applicable in the conceptual design of a sports competition hall with the aid of the new Gh-mF node. It is particularly useful for the relatively late sub-phase of the conceptual design where the main purpose is to reduce existing design possibilities. Thanks to the one-time Optimization Problem Re-Formulation (Re-OPF) process, this subtype method can help achieve a better Pareto front in terms of the quantity, pertinence, proximity, and diversity, and geometric variation appropriateness, compared with adopting the traditional method (defined in FIG.3.1). These benefits can be affected by factors like over-screening objectives, choosing different high-level variable values (i.e., "*RoofSteps*" values), over-screening low-level variables, and utilizing a purely random initial population.

References

Al-Obaidi, K. M., & Rahman, A. M. A. (2016). Toplighting systems for improving indoor environment: a review. In: M. Ahmad, M. Ismail, & S. Riffat (Eds.), Renewable Energy and Sustainable Technologies for Building and Environmental Applications (pp. 117-136). Springer, Cham. ISBN: 978-3-319-31838-7.

Beltran, L. O. (2005). Toplighting for the Tropics. In: D. K. Raydan, H. K. Melki (Eds.), Environmental Sustainability: The Challenge of Awareness in Developing Societies (pp. 297-300). NDU Press, Lebanon. ISBN: 9953418691.

Big ladder (2021). Group - zone forced air units. https://bigladdersoftware.com/epx/docs/8-0/input-output-reference/page-032.html [Accessed: 24-September-2021].

CIBSE (1999). Daylighting and Window Design. The Chartered Institute of Building Service Engineers (CIBSE), London. ISBN: 0900953985.

DCMS (2008). Guide to Safety at Sports Grounds. Fifth Edition. The Stationery Office, London. ISBN: 978-0-11-702074-0.

Didoné, E. L., & Pereira, F. O. R. (2011). Integrated computer simulation for considering daylight when assessing energy efficiency in buildings. In: Proceedings of Building Simulation 2011 - 12th Conference of International Building Performance Simulation Association (pp. 2102-2109). IBPSA. ISSN: 2522-2708.

Futrell, B. J., Ozelkan, E. C., & Brentrup, D. (2015). Bi-objective optimization of building enclosure design for thermal and lighting performance. Building and Environment, 92, 591-602. DOI: https://doi.org/10.1016/j.buildenv.2015.03.039.

Glen, S. (2021). Correlation Coefficient: Simple Definition, Formula, Easy Steps. https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/ [Accessed: 24-September-2021].

Gu, C. (2013). Smoothing Spline ANOVA Models. Springer, New York. ISBN: 978-0-387-95353-3.

Harntaweegonsa, S., & Beltran, L. (2007). Daylighting Performance of Toplighting Systems in the Hot and Humid Climate of Thailand. In: Solar 2006: Renewable Energy - Key to Climate Recovery (pp. 613-618). Curran Associates, New York. ISBN: 978-1604232882.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323. DOI: https://doi.org/10.1145/331499.331504.

Lechner, N. (2014). Heating, Cooling, Lighting: Sustainable Design Methods for Architects. Fourth Edition. John Wiley & Sons, Hoboken. ISBN: 978-1-118-58242-8.

Mavridou, T., & Doulos, L. T. (2019). Evaluation of different roof types concerning daylight in industrial buildings during the initial design phase: Methodology and case study. Buildings, 9(7), 170. DOI: https://doi.org/10.3390/buildings9070170.

Nabil, A., & Mardaljevic, J. (2006). Useful daylight illuminances: A replacement for daylight factors. Energy and buildings, 38(7), 905-913. DOI: https://doi.org/10.1016/j.enbuild.2006.03.013.

Rahman, A. S. (2019). Effects of nanofibers on properties of geopolymer composites. In: F. Pacheco-Torgal, M. V. Diamanti, A. Nazari, C. G. Granqvist, A. Pruna, & S. Amirkhanian (Eds.), Nanotechnology in Eco-efficient Construction (pp. 123-140). Woodhead Publishing, Sawston. ISBN: 978-0-08-102641-0.

Ramos, G., & Ghisi, E. (2010). Analysis of daylight calculated using the EnergyPlus programme. Renewable and Sustainable Energy Reviews, 14(7), 1948-1958. DOI: https://doi.org/10.1016/j.rser.2010.03.040.

Ricco, L., Rigoni, E., & Turco, A. (2013). Smoothing spline ANOVA for variable screening. Dolomites Research Notes on Approximation, 6 (Special_Issue), 130-139. DOI: https://doi.org/10.14658/pupj-drna-2013-Special_Issue-13.

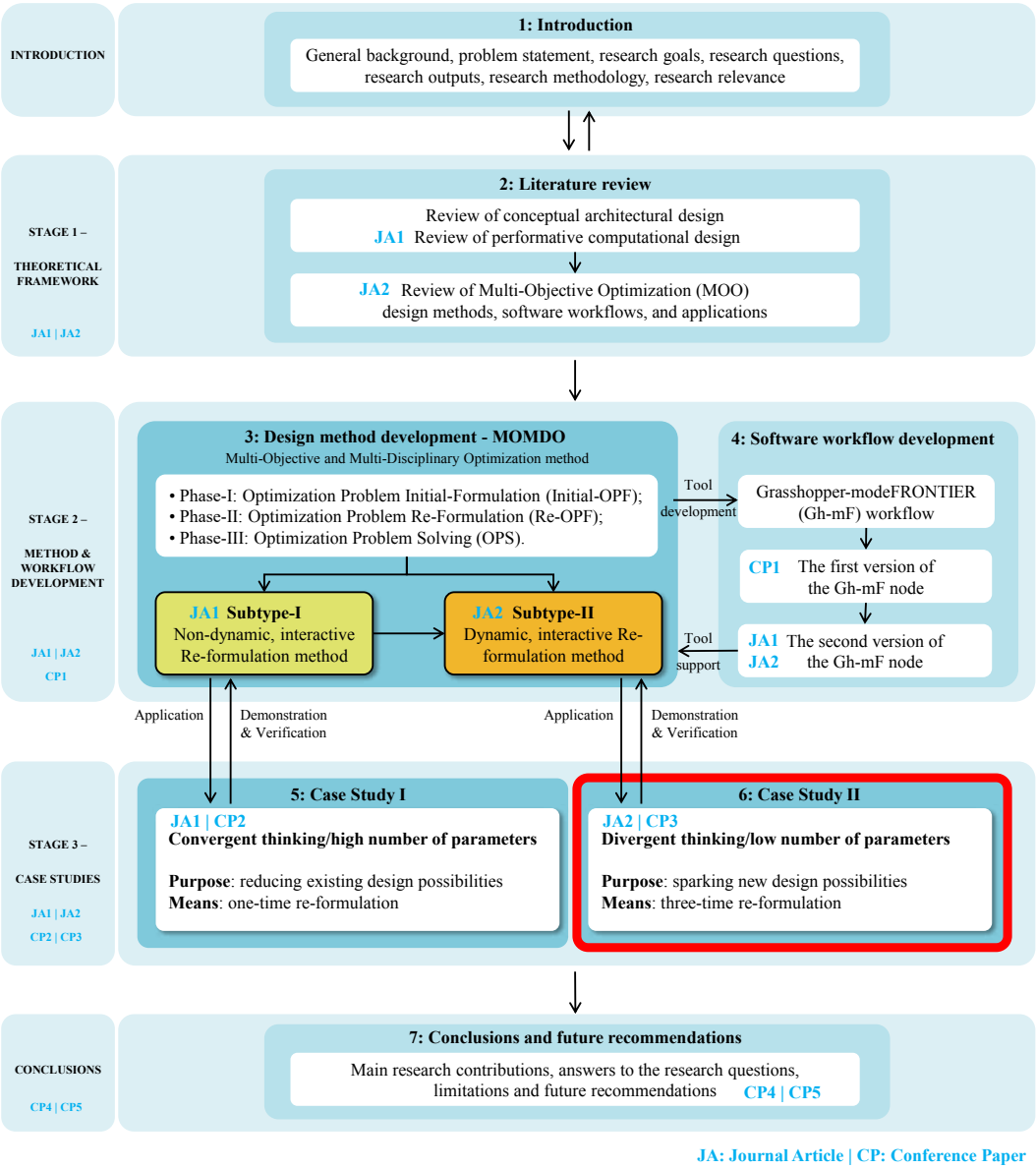
Sun, Y., Xiong, L., & Su, P. (2013). Grandstand Grammar and its Computer Implementation. In: R. Stouffs, & S. Sariyildiz (Eds.), Computation and Performance: Proceedings of the 31st Conference on Education in Computer Aided Architectural Design in Europe, Volume 2 (pp. 645-654). eCAADe and Delft University of Technology. ISBN: 978-94-91207-05-1.

Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. Automation in Construction, 92, 242-269. DOI: https://doi.org/10.1016/j.autcon.2018.03.023.

Yang, D., Turrin, M., Sariyildiz, S., & Sun, Y. (2015). Sports building envelope optimization using multi-objective multidisciplinary design optimization (M-MDO) techniques: Case of indoor sports building project in China. In: 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 2269-2278). IEEE, Piscataway. ISBN: 978-1-4799-7492-4.

Zitzler, E., Brockhoff, D., & Thiele, L. (2007). 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 (pp. 862-876). Springer, Berlin, Heidelberg. ISBN: 978-3-540-70927-5.

Zitzler, E., & Thiele, L. (1998). 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 (pp. 292-301). Springer, Berlin, Heidelberg. ISBN: 978-3-540-65078-2.


INTRODUCTION
1: Introduction
General background, problem statement, research goals, research questions, research outputs, research methodology, research relevance
STAGE 1 – THEORETICAL FRAMEWORK
JA1 | JA2
2: Literature review
Review of conceptual architectural design
JA1 Review of performative computational design
JA2 Review of Multi-Objective Optimization (MOO) design methods, software workflows, and applications
STAGE 2 – METHOD & WORKFLOW DEVELOPMENT
JA1 | JA2
CP1
3: Design method development - MOMDO
Multi-Objective and Multi-Disciplinary Optimization method
• Phase-I: Optimization Problem Initial-Formulation (Initial-OPF);
• Phase-II: Optimization Problem Re-Formulation (Re-OPF);
• Phase-III: Optimization Problem Solving (OPS).
JA1 Subtype-I
Non-dynamic, interactive Re-formulation method
JA2 Subtype-II
Dynamic, interactive Re-formulation method
Tool development
4: Software workflow development
Grasshopper-modeFRONTIER (Gh-mF) workflow
CP1 The first version of the Gh-mF node
JA1 JA2 The second version of the Gh-mF node
Tool support
Application
Demonstration & Verification
Application
Demonstration & Verification
STAGE 3 – CASE STUDIES
JA1 | JA2
CP2 | CP3
5: Case Study I
JA1 | CP2
Convergent thinking/high number of parameters
Purpose: reducing existing design possibilities
Means: one-time re-formulation
6: Case Study II
JA2 | CP3
Divergent thinking/low number of parameters
Purpose: sparking new design possibilities
Means: three-time re-formulation
CONCLUSIONS
CP4 | CP5
7: Conclusions and future recommendations
Main research contributions, answers to the research questions, limitations and future recommendations CP4 | CP5
JA: Journal Article | CP: Conference Paper

TOC

6 Case Study II

This chapter presents Case Study II. In this case study, the Subtype-II method (i.e., dynamic method) is applied to the conceptual design of the skylight geometry of a sports training hall with the aid of the new Gh-mF node. This case study was selected to demonstrate and verify this subtype method primarily because it focused on the relatively early sub-phase of the conceptual design where divergent thinking is often highlighted or the number of parameters is usually low.

The chapter is structured as follows. First, it introduces the purpose of Case Study II (Section 6.1). Then, it provides the background of the project involved in this case study (Section 6.2). Next, it presents the results derived from each phase of the dynamic method (Section 6.3, 6.4, 6.5). Finally, it concludes by summarizing the main research results, identifying possible applications of the dynamic method, and providing concluding remarks (Section 6.6).

This case study was supported by ESTECO SpA. Ir. Danilo Di Stefano, the product manager, who provided great support during the use of modeFRONTIER, especially the use of the Self-Organizing Map. Sections 6.2-6.5 involve contents published in Journal Article 2 (Yang et al., 2020) and Conference Paper 3 (Yang et al., 2017).

6.1 Introduction

The purposes of Case Study II are multifold. First of all, this case study demonstrates the use of the Subtype-II method (i.e., dynamic method). Second, it verifies the benefits of adopting the method and the factors affecting the behaviors of the method. Third, it provides valuable feedback for possible applications of the method.

This case study assumes that the design context is to highlight sparking new design possibilities, such as many circumstances in the relatively early sub-phase of the conceptual design. Thus, the Subtype-II method (i.e., dynamic method) is adopted. This subtype method contains three phases; and the re-formulation phase is cyclical, as illustrated in FIG.6.1.

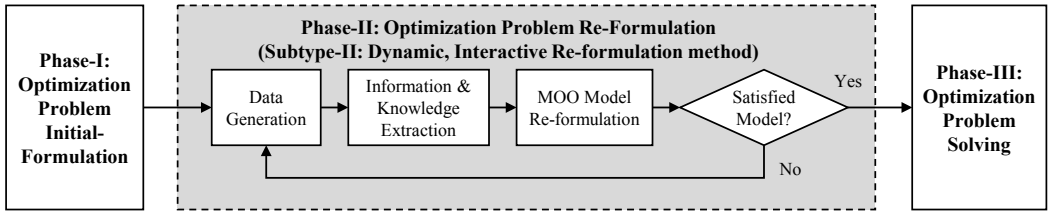


FIG. 6.1 The scheme of applying the Subtype-II method

Note: The shaded region corresponds to FIG.3.5.

6.2 Project description

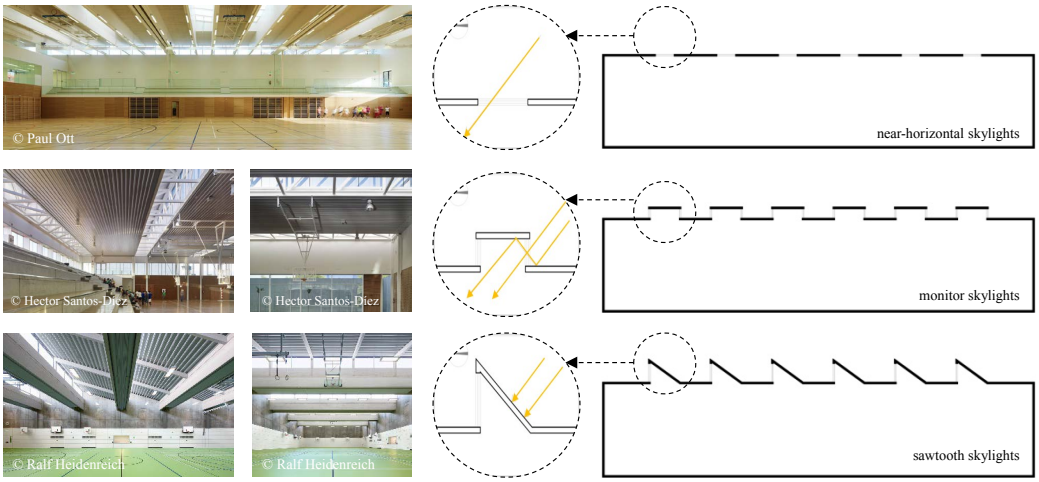


FIG. 6.2 A cuboid shape gymnasium and related examples (revised from Yang et al., 2017)

Note: A gymnasium with near-horizontal skylights (top); with monitor skylights (middle); with sawtooth skylights (bottom). Image Source: https://www.archdaily.com/.

This case study is based on a hypothetical project - a cuboid shape gymnasium (see FIG.6.2). The project is a simplified example of indoor sports halls that utilize top daylighting; and it is designed by the author for investigating different types of rooftop daylighting systems for indoor sports halls.

TOC

The site of the project is located in Guangzhou, China. According to Chinese building codes – Code for thermal design of civil building (GB 50176-2016) and Standard for daylighting design of buildings (GB 50033-2013), the project is in the Hot Summer and Warm Winter climate zone and the IV daylighting climate zone.

The project contains a training hall without grandstands. The size of the court is 40m × 70m, which meets the requirements of many dry sports activities (e.g., basketball, badminton, gymnastics). This case study manipulates the geometries of the roofs, skylights, and internal shadings of the hall, in order to meet architectural, daylighting, energy, and cost performances.

6.3 Phase-I involving multiple initial concepts

Phase-I (i.e., Optimization Problem Initial-Formulation) of the Subtype-II method (i.e., dynamic method) involves multiple initial concepts.

In Case Study II, the initial concepts are three typical top daylighting concepts (Section 6.3.1). Based on these concepts, a geometric parametric model and multi-disciplinary simulation or calculation models are created and integrated (Section 6.3.2), thus formulating an initial MOO model as the main output of this phase (Section 6.3.3).

6.3.1 Initial concept generation

The initial concepts are three typical top daylighting concepts proposed at the outset of the conceptual design (see FIG.6.3). These concepts are near-horizontal skylights (i.e., Concept 1_0), monitor skylights (i.e., Concept 2_0), and sawtooth skylights (i.e., Concept 3_0), which utilize different control strategies to manage the quantity and quality of constantly changing daylight (CIBSE, 1999; Beltran, 2005; Harntaweegonsa and Beltran, 2007; Lechner, 2014; Al-Obaidi and Rahman, 2016; Mavridou and Doulos, 2019). They are chosen for discussion, because of their ability to bring natural lights deep into buildings.

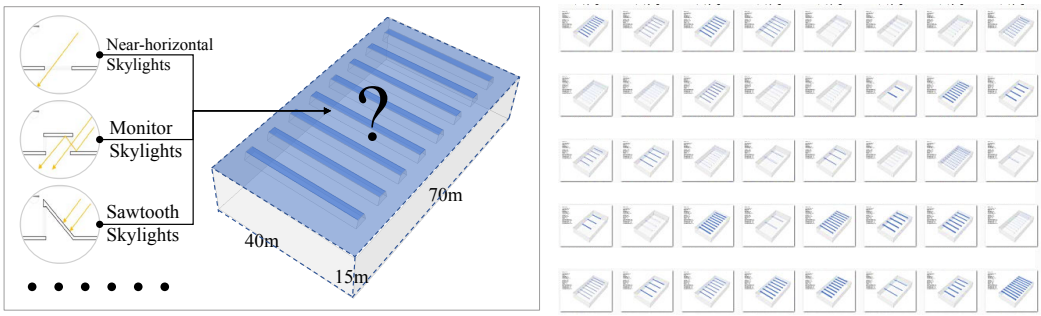


FIG. 6.3 The initial concept in Case Study II (revised from Yang et al., 2020)

Note: Three typical top daylighting concepts (left), possible geometric variations generated based on these concepts (right).

6.3.2 Multi-objective optimization model initial formulation

6.3.2.1 Geometric parametrization

A geometric parametric model is created, based on the three initial top daylighting concepts illustrated in FIG.6.4 (1-3). Each of the concepts implies a vast number of possible building geometries. The complexity level of the geometries is not very high, as the focus of this case is not on highlighting the complexity of the geometries, but rather on showing how to continually enrich concepts in an informed manner. The geometries are initially parameterized by the design variables shown in TABLE 6.1 (1-3). Given that all lighting bands in a concept are the same, only one of them is parametrically defined. The design variables are organized in a two-level hierarchical structure, to facilitate the geometric parameterization of the three initial concepts. More details are provided below.

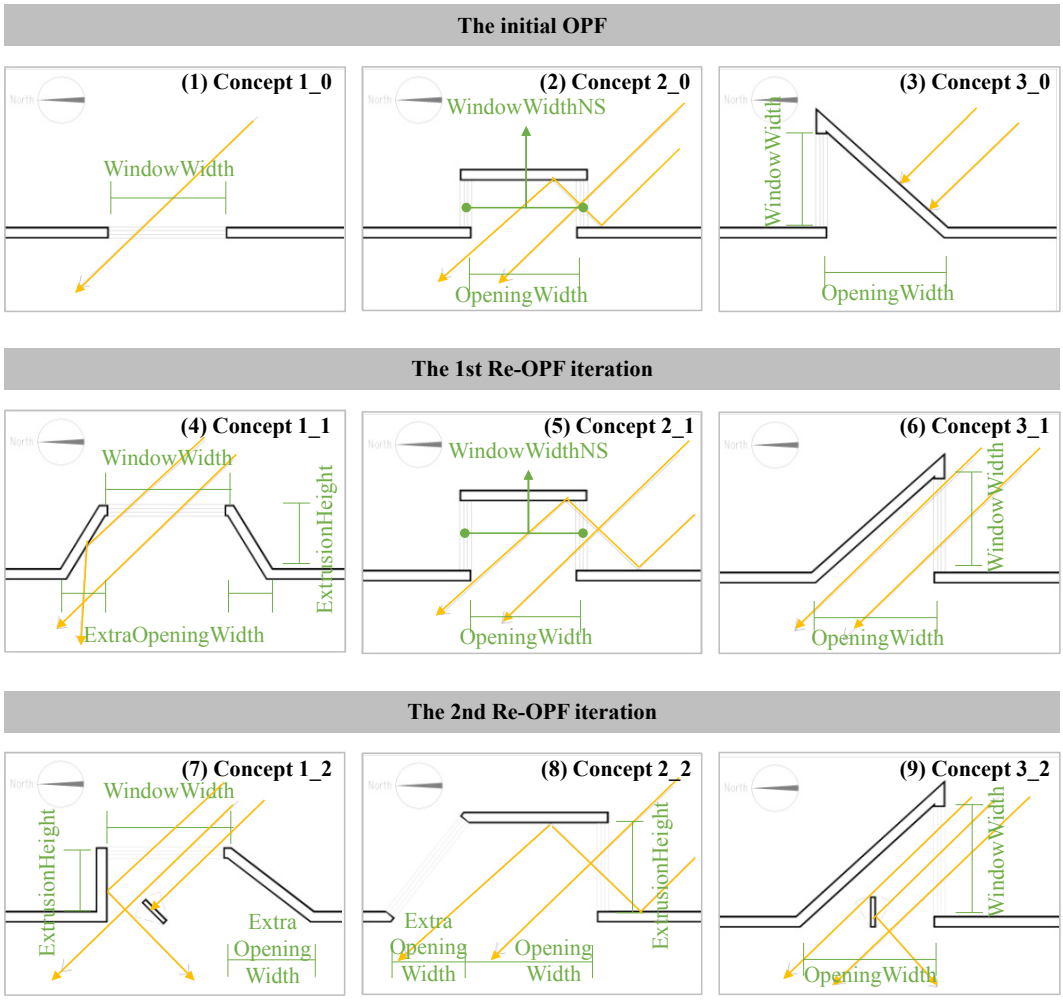


FIG. 6.4 The initial and consequent parametric models in Case Study II (revised from Yang et al., 2020)

Note: The high-level variable is "Concept" which defines different concepts. The initial parametric model that defines Concept 1_0, Concept 2_0, Concept 3_0 in the initial formulation phase (1-3); the revised parametric model that defines Concept 1_1, Concept 2_1, Concept 3_1 in the first re-formulation iteration (4-6); the revised parametric model that defines Concept 1_2, Concept 2_2, Concept 3_2 in the second re-formulation iteration (7-9).

TABLE 6.1 The initial and consequent design variables in Case Study II (revised from Yang et al., 2020)

Design Variables	Range	Step	Design Variables	Range	Step	Design Variables	Range	Step
Concept	1-3	1	Concept	1-3	1	Concept	1-3	1
LightingBandNumber	2-10	1	LightingBandNumber	2-10	1	LightingBandNumber	2-10	1
WindowLength	30-38m	0.1m	WindowLength	30-38m	0.1m	WindowLength	30-38m	0.1m
WindowWidth	1-3m	0.1m	WindowWidthNS	1-3m	0.1m	WindowWidth	1-3m	0.1m
			OpeningWidth	1-3m	0.1m	OpeningWidth	1-3m	0.1m
(1) Concept 1_0			(2) Concept 2_0			(3) Concept 3_0		
Concept	4-6	1	Concept	4-6	1	Concept	4-6	1
LightingBandNumber	2-10	1	LightingBandNumber	2-10	1	LightingBandNumber	2-10	1
WindowLength	30-38m	0.1m	WindowLength	30-38m	0.1m	WindowLength	30-38m	0.1m
WindowWidth	**2-4m**	**0.1m**	**WindowWidthNS**	**2-4m**	**0.1m**	**WindowWidth**	**2-4m**	**0.1m**
ExtrusionHeight	**1-3m**	**0.1m**	OpeningWidth	1-3m	0.1m	OpeningWidth	1-3m	0.1m
ExtraOpeningWidth	**1-3m**	**0.1m**				**BuildingOrientation**	**165-195**	**0.5**
(4) Concept 1_1			(5) Concept 2_1			(6) Concept 3_1		
Concept	7-9	1	Concept	7-9	1	Concept	7-9	1
LightingBandNumber	2-10	1	LightingBandNumber	2-10	1	LightingBandNumber	2-10	1
WindowLength	30-38m	0.1m	WindowLength	30-38m	0.1m	WindowLength	30-38m	0.1m
WindowWidth	**2-4m**	**0.1m**	OpeningWidth	2-4m	0.1m	**WindowWidth**	**2-4m**	**0.1m**
ExtrusionHeight	**1-3m**	**0.1m**	ExtrusionHeight	1-3m	0.1m	OpeningWidth	2-4m	0.1m
ExtraOpeningWidth	**1-3m**	**0.1m**	ExtraOpeningWidth	1-3m	0.1m	**BuildingOrientation**	**165-195**	**0.5**
ShadingAngleA	15-75	0.5				ShadingAngleB	60-120	0.5
(7) Concept 1_2			(8) Concept 2_2			(9) Concept 3_2		

Note: The variables in dark green and light green cells respectively represent the high-level variable and the low-level variables. The bolded black texts in the light green cells represent the newly added and revised variables during the first re-formulation interation, while the bolded green texts in the light green cells represent the newly added and revised variables during the second re-formulation iterations.

Geometric parameterization of the near-horizontal skylights

The geometric parameterization of the near-horizontal skylights specifically refers to the creation of parametric schemata which define the geometries of near-horizontal rooftop windows (i.e., Concept 1_0).

For defining the near-horizontal skylights, the variables LightingBandNumber, WindowLength, and WindowWidth are used, as illustrated in FIG.6.4 (1). The first variable represents the number of the lighting bands in the concept. The last two variables respectively represent the length and width of a window in the concept.

Geometric parameterization of the monitor skylights

The geometric parameterization of the monitor skylights specifically refers to the creation of parametric schemata which define the geometries of vertical two-sided rooftop windows (i.e., Concept 2_0).

For defining the monitor skylights, the variables LightingBandNumber, WindowLength, WindowWidthNS, and OpeningWidth are used, as illustrated in FIG.6.4 (2). The first two variables are the same as those used for defining the near-horizontal skylights. The third variable represents the sum of the widths of the north- and south-facing windows in a single lighting band. The fourth variable represents the width of the roof opening in a single lighting band.

Geometric parameterization of the sawtooth skylights

The geometric parameterization of the sawtooth skylights specifically refers to the creation of parametric schemata which define the geometries of vertical one-sided rooftop windows (i.e., Concept 3_0).

For defining the sawtooth skylights, the variables LightingBandNumber, WindowLength, WindowWidth, and OpeningWidth are used, as illustrated in FIG.6.4 (3). The first three variables are the same as those used for defining the near-horizontal skylights. The last variable is the same as that used for defining the monitor skylights.

The bounds and intervals of the variables are tuned based on some daylighting rules of thumb, to avoid unfeasible design solutions, while maintaining rich variability of the concepts.

A two-level hierarchical variable structure is used to facilitate the exploration of multiple concepts simultaneously. The high-level variable is the variable "*Concept*" marked in dark green in TABLE 6.1 (1-3) and is used to label different concepts under consideration. The low-level variables are those marked in light green in TABLE 6.1 (1-3) and are used to define the geometries of these concepts. When the value of the "*Concept*" changes, a different set of low-level variables are selected automatically to define the geometries of the associated concept. In this way, it is convenient to switch among the geometries of the concepts.

6.3.2.2 Simulation integration

Multi-disciplinary simulation or calculation models are integrated with the geometric parametric model. The initial concepts are meant for meeting a set of multi-disciplinary performance requirements, including daylight, energy, and cost requirements. The completeness level of the requirements is not very high, as the focus of this case is not on discussing the completeness of requirements, but rather on showing how to find meaningful measures in an informed manner. The requirements are initially represented by the performance measures shown in TABLE 6.2. There can be multiple performance measures for the same kind of performance requirement. They can be considered as objectives or constraints. More details are provided below.

TABLE 6.2 The initial performance measures in Case Study II (Yang et al., 2020)

Disciplines	Performance Measures	Objectives	Abbreviations	Definitions
Energy use	Energy Use Intensity	↓	EUI	Energy used per square meter of floor area
	Percentages of Cooling	↓	PoC	Percentages of energy used for cooling, heating, lighting, and equipment, respectively (which can be used as meaningful objectives, if they account for major portions of energy use)
	Percentages of Heating	↓	PoH	
	Percentages of Lighting	↓	PoL	
	Percentages of Equipment	↓	PoE	
Daylight availability	Useful Daylight Illuminance (<100)	↓	UDI (<100)	Percentage of floor area that meets the specified illuminance range for at least 50% of the occupied time (i.e., UDI_{mod})
	Useful Daylight Illuminance (100-2000)	↑	UDI (100-2000)	
	Useful Daylight Illuminance (>2000)	↓	UDI (>2000)	
	Day Lit Area	↑	DLA	Percentage of floor area that receives illuminances above 300 lx for at least 50% of the occupied time (i.e., DLA_{mod})
	Over Lit Area	↓	OLA	Percentage of floor area that receives illuminances above 3000 lx for at least 5% of the occupied time (i.e., OLA_{mod})
Daylight uniformity	Average Uniformity	↑	AU	Annual average of illuminance uniformity ratios (i.e., UR_{mod})
Cost for glass	Area of Glass	↓	AoG	Total area of the glass used for top windows

Integration of climatic simulation

To obtain climatic performance feedback, a daylight simulation model and an energy simulation model are created and coupled. The inputs (i.e., geometries and parameters) and outputs (i.e., performance measures) of these models, and the software tools used for creating these models are specified below.

First, the input geometries of the simulation models include the parameterized geometries of the rooftop windows (described in Section 6.3.2.1) and the non-parameterized geometry of the 40m × 70m × 15m hall in question. The rooftop windows are the only spots through which daylight and solar heat gain are received.

Second, the input parameters of the simulation models are mostly the same as those used in Case Study I, except the weather file. In this case here, the weather file of Guangzhou derived from Chinese Standard Weather Data is used.

Third, the output performance measures of the simulation models include those used in Case Study I, and some other possible measures. All these measures are initially treated as optimization goals.

For measuring daylight availability, the performance measure UDI_{mod} defined in Case Study I is also used in this case. The range of "*useful*" daylight illuminances is still 100-2000 lux, thus the measure UDI_{mod} here is denoted by UDI (100-2000). The range of "*useless*" daylight illuminances can be smaller than 100 lux or larger than 2000 lux, thus other possible measures include Useless Daylight Illuminance (<100) denoted by UDI (<100), or Useless Daylight Illuminance (>2000) denoted by UDI (>2000). What further differs from Case Study I is that, in this case, the percentage of occupied hours for which the "*useful*" or "*useless*" daylight illuminances are received is set to 50%. Moreover, Day Lit Area denoted by DLA and Over Lit Area denoted by OLA are also possible measures. They are respectively defined as the percentage of floor area that receives illuminances above 300 lux for at least 50% of the occupied time, and the percentage of floor area that receives illuminances above 3000 lux for at least 5% of the occupied time. The UDI (100-2000) and DLA are to be maximized; the UDI (<100), UDI (>2000), and OLA are to be minimized.

For measuring daylight uniformity, the performance measure UR_{mod} defined in Case Study I is also used in this case. Given that the UR_{mod} is actually the annual average of illuminance uniformity ratios, it is called Average Uniformity and denoted by AU here. The AU is to be maximized.

For measuring energy usage, the performance measure EUI defined in Case Study I is also used in this case. Moreover, Percentages of Lighting (PoL), Percentages of Cooling (PoC), Percentages of Heating (PoH), and Percentages of Equipment (PoE) are also possible measures, which are respectively defined as the percentage of energy use for lighting, cooling, heating, and equipment. The EUI, PoL, PoC, PoH, and PoE are to be minimized.

Finally, the software tools used for creating the simulation models are Grasshopper's plug-ins called Ladybug and Honeybee which adopt Daysim and EnergyPlus simulation engines, as in Case Study I.

Integration of cost calculation

To obtain cost performance feedback, a simple cost calculation model is created. The inputs (i.e., geometries and parameters) and outputs (i.e., performance measures) of this model and the software tools used for creating this model are specified below.

First, the input geometries of the calculation model include the parameterized geometries of the near-horizontal skylights, monitor skylights, and sawtooth skylights (described in Section 6.3.2.1). These parametrically changeable geometries determine the amount of glass used for the rooftop windows, and hence affect the cost calculations.

Second, the input parameters of the calculation model can be omitted in this case. Specifically, the glass unit price can be omitted because the same type of glass is used in this case.

Third, the output performance measures of the calculation model include the Area of Glass (AoG). This measure is initially treated as an optimization goal.

Reducing glass costs is important for indoor sports halls to save initial investment costs. The AoG is a performance measure useful for reflecting the glass cost. It represents the total area of glass used for the rooftop windows and is proportional to glass cost. The AoG is to be minimized.

Finally, the software tools used for creating the calculation model are Grasshopper's native components. These components are used to facilitate the geometric parameterization and hence the cost calculations (i.e., glass area calculations).

6.3.3 Output of Phase-I

The main output of Phase-I is an initial MOO model which includes an initial set of performance objectives, constraints, and design variables. This model is to be used in the next phase.

6.4 Phase-II adopting a cyclical re-formulation process

Phase-II (i.e., Optimization Problem Re-Formulation) of the Subtype-II method (i.e., dynamic method) adopts a cyclical process.

In Case Study II, the design context is to highlight sparking new design possibilities; thus, the cyclical re-formulation process adopted is a three-time re-formulation process that focuses on adding new variables (i.e., enriching new concepts divergently). For this re-formulation, designers have to discover the answers to the following questions:

- Which performance measures are more meaningful for final objectives or constraints?
- Which existing concepts are more promising to lead to good quantitative and qualitative performances?
- Which new concepts are more promising to lead to good quantitative and qualitative performances?
- How to achieve a proper MOO model that has the potential to lead to better Pareto solutions?

Normally, designers answer these questions highly relying on their personal past experiences. However, with the increase of the performance objectives, constraints, and design variables, the complexity of these questions increases rapidly. In this circumstance, it is difficult for designers to obtain the right answers to these questions by purely relying on their past experiences.

Given the above fact, it is necessary to carry out quantitative data analysis from multiple angles in this phase, in order to extract useful information and knowledge and to support the cyclical re-formulation process. As exemplified in FIG.3.5, three types of actions are conducted in each re-formulation iteration (Section 6.4.1, 6.4.2 and 6.4.3) for achieving a final MOO model (Section 6.4.4).

6.4.1 The first-time re-formulation

6.4.1.1 Data generation

The first group of 300 data sets is generated for analysis. It is derived based on an automation process consisting of a MOO model, the Uniform Latin Hypercube sampling algorithm, and the sequential execution order (see FIG.6.5). A data set contains quantitative data (i.e., numeric design values and performance values) and qualitative data (i.e., building geometries).

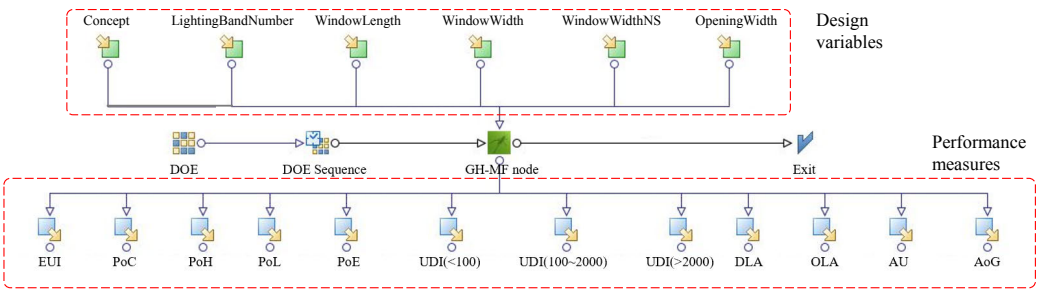


FIG. 6.5 The automation process for generating the data in Case Study II

Note: The variables in the automation process can be revised for each re-formulation iteration.

6.4.1.2 Information and knowledge extraction

Based on the first group of 300 data sets, three categories of knowledge are extracted, namely knowledge about the performance measures, promising existing concepts, and promising new concepts, as exemplified in FIG.3.5.

Knowledge about the performance measures

To acquire knowledge about which quantitative performance measures are more meaningful for final objectives or constraints, it is helpful to extract correlations between related measures. The quantitative data (having all the initial measures as its dimensions) is analyzed using a specific correlation analysis technique – Self-Organizing Map (Kohonen, 2001). As a result, SOM planes of all the initial measures are generated and visualized on a large hexagonal grid (see FIG.6.6, left). Based on the result, correlation information is extracted and interpreted, so as to acquire desired knowledge, as described below.

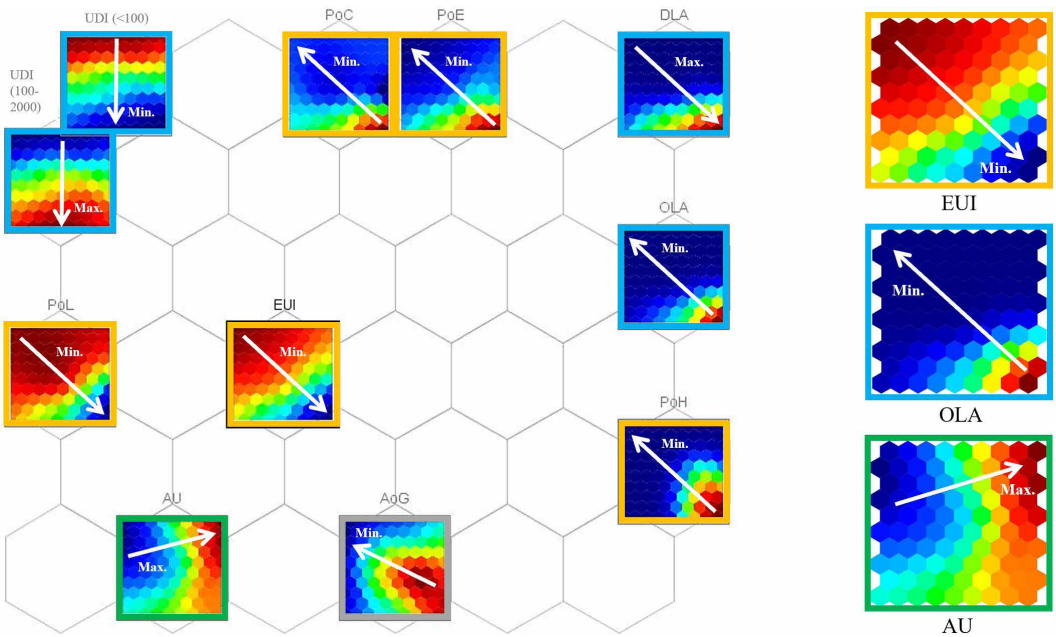


FIG. 6.6 Self-Organizing Map planes (Yang et al., 2020)

Note: SOM planes of all the initial measures (left); SOM planes of the three chosen quantitative performance measures (right). The yellow, blue, green and grey boxed lines show SOM planes of energy use, daylight availability, daylight uniformity and investment measures, respectively.

1 Regarding EUI and PoL, EUI is a meaningful quantitative measure, as it can facilitate a direct comparison of energy use among different buildings; PoL is also a meaningful quantitative measure, as the lighting energy use, in this case, dominates the total energy use. Given that they are positively and strongly correlated, and their optimization goals are the same, one of them can be removed.

2 Regarding UDI (<100) and UDI (100-2000), UDI (<100) is a meaningful quantitative measure for reducing "*useless*" daylight illuminances; UDI (100-2000) is a meaningful quantitative measure for increasing "*useful*" daylight illuminances. Given that they are negatively and strongly correlated, and their optimization goals are opposite, one of them can be removed.

3 Regarding OLA and DLA, OLA is a meaningful quantitative measure that not only reflects daylight availability but also glare or overheating risk; DLA is a redundant quantitative measure, as it is negatively and considerably correlated with EUI (and PoL) and has an opposite optimization goal. Given these facts, DLA can be removed.

4 Regarding AU and AoG, they are meaningful quantitative measures in different senses. Given that they have weak correlations with the above-mentioned measures, either of them can be kept.

To acquire knowledge about which qualitative performance measures are more meaningful to be chosen to form final objectives or constraints, human subjectivity is required (i.e., subjectively determining meaningful qualitative measures). A qualitative measure Aesthetics can be considered more meaningful to be chosen to form a constraint, among many possible measures (e.g., cultural, social related measures). It is defined as the aesthetic quality of rooftop windows specifically.

Overall, in this case, the quantitative measures EUI, OLA, and AU are considered more meaningful to be chosen to form final objectives (i.e., Min_EUI, Min_OLA, Max_AU, as shown in FIG.6.6, right); and the qualitative measure Aesthetics is considered more meaningful to be chosen to form a final constraint.

Knowledge about promising existing concepts

To acquire knowledge about which existing concepts are more promising to lead to good EUI, OLA, and AU performances, it is helpful to construct meaningful clusters of samples according to "*Concept*" values and performance values. The quantitative data (having the high-level variable "*Concept*" and measures EUI, OLA, and AU as its dimensions) is analyzed using a specific clustering analysis technique – Hierarchical Clustering (Jain et al., 1999). As a result, nine clusters of samples are generated and visualized in a parallel coordinate chart (see FIG.6.7, left). The clustering information is interpreted, so as to form useful knowledge, as described below.

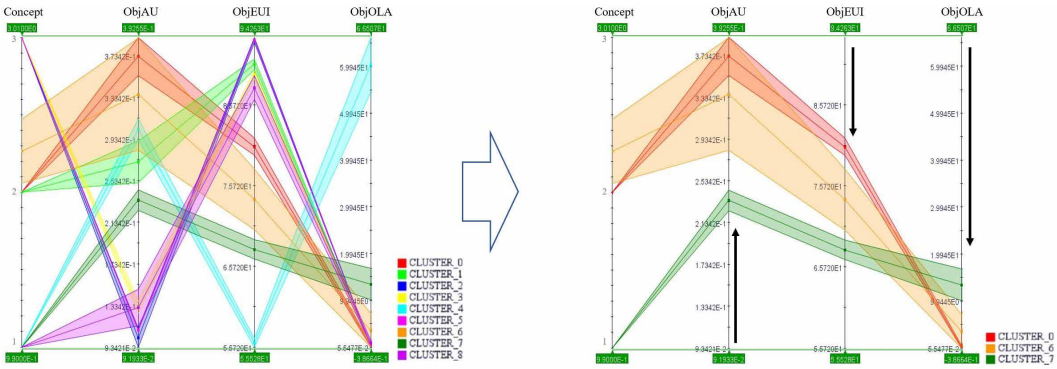


FIG. 6.7 Clusters in the first re-formulation iteration (Yang et al., 2020)

Note: 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.

1. Samples in CLUSTER_0 belonging to Concept 2_0, samples in CLUSTER_6 mostly belonging to Concept 2_0, and samples in CLUSTER_7 belonging to Concept 1_0, can reach the chosen objectives simultaneously (see FIG.6.7, right).

2. Concept 1_0 and Concept 2_0 can be considered as quantitatively more promising existing concepts, given that the identified clusters of samples mostly belong to these two concepts.

3. Concept 3_0 can be considered as a quantitatively less promising existing concept, given that only very few identified samples in CLUSTER_6 belong to this concept.

To acquire knowledge about which existing concepts are more promising to lead to acceptable Aesthetics performance, human subjectivity is required. Here, it is assumed that Concept 1_0 and Concept 2_0 have less promising Aesthetics performance, Concept 3_0 has more promising Aesthetics performance.

Overall, in this stage, the existing Concept 1_0 and Concept 2_0 are considered quantitatively more promising and qualitatively acceptable; the existing Concept 3_0 is considered qualitatively more promising and quantitatively acceptable.

Knowledge about promising new concepts

To acquire knowledge about which new concepts are more promising to lead to good EUI, OLA, and AU performances, it is helpful to understand quantitative performance distributions of the existing concepts. The quantitative data (having the measures EUI, OLA, and AU as its dimensions) is analyzed using statistical summary techniques. As a result, the quantitative performance distributions of Concept 1_0, Concept 2_0, Concept 3_0, are visualized in box-whisker plots and scatter plots (see FIG.6.8). The distribution information is interpreted, so as to form useful knowledge, as described below.

1 Regarding Concept 1_0, there is room for possible improvements of OLA and AU, while maintaining EUI. OLA and AU can be improved by introducing more reflected daylight. EUI can be maintained by keeping daylight entering through the top-facing windows without obstacles. Concept 1_1 is created based on these strategies - by lifting the skylights, enlarging the window size and the openings on the roof bottom surface (i.e., creating inclined opaque elements).

2 Regarding Concept 2_0, there is room for possible improvement of EUI, while maintaining OLA and AU. EUI can be improved by introducing more daylight into the space. OLA and AU can be maintained by blocking out daylight from high angles with the horizontal opaque elements. Concept 2_1 is created based on these strategies - by lifting the protruding elements (i.e., enlarging the window size).

3 Regarding Concept 3_0, there is room for possible improvements of EUI and AU, while maintaining OLA. EUI and AU can be improved by introducing slightly more daylight from around the south. OLA can be maintained by blocking out daylight from the opposite side of the windows with the inclined opaque elements. Concept 3_1 is created based on these strategies - by changing the orientation of the window to the south and slightly enlarging the window size.

To acquire knowledge about which new concepts are more promising to lead to acceptable Aesthetics performance, human subjectivity is required. Here, it is assumed that Concept 1_1 and Concept 2_1 have less promising Aesthetics performance, Concept 3_1 has more promising Aesthetics performance, and that no other new concepts are generated only for an aesthetical purpose.

Overall, in this stage, the new Concept 1_1, Concept 2_1, and Concept 3_1 are considered helpful for improving quantitative performances and maintaining acceptable qualitative performance.

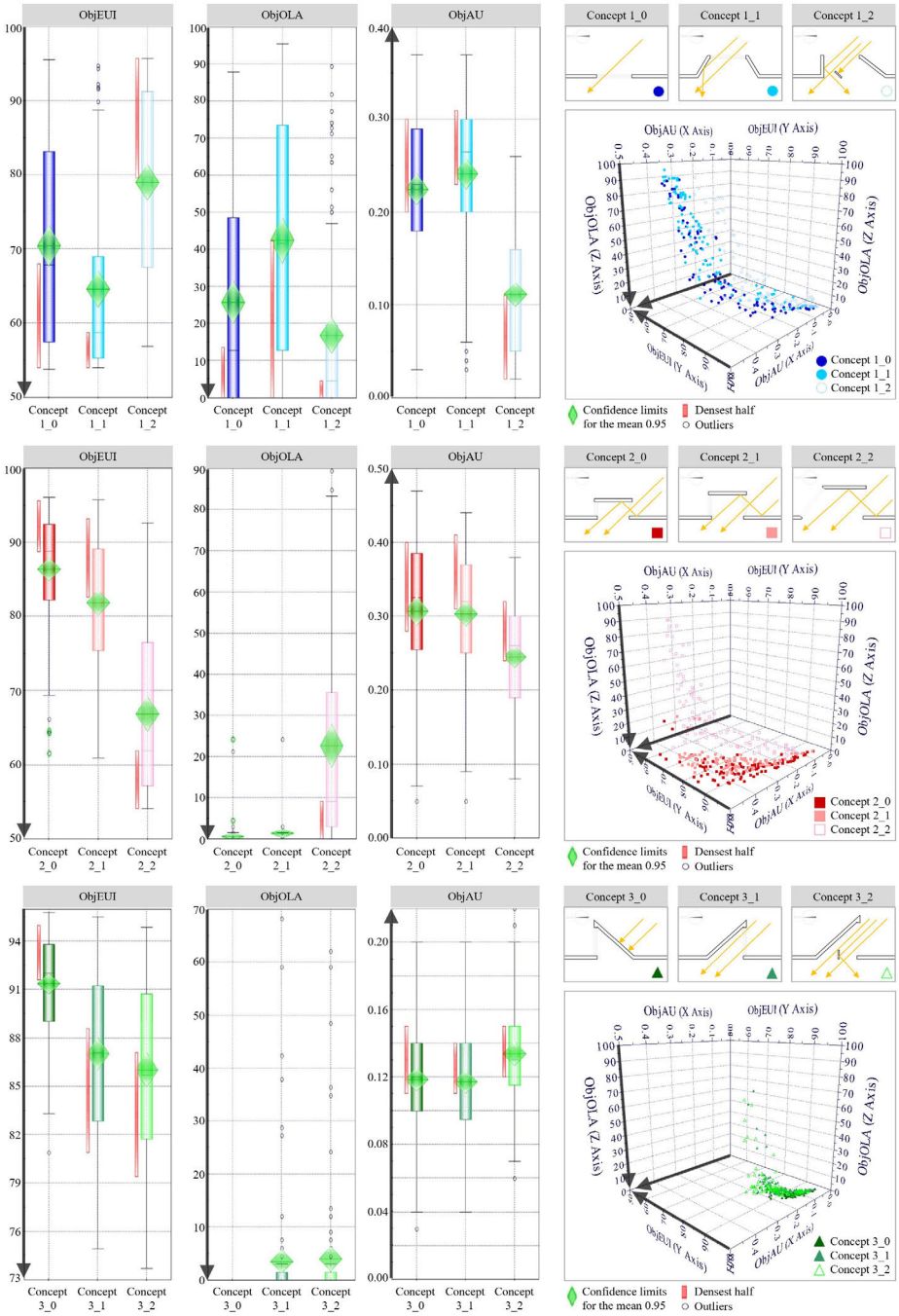


FIG. 6.8 Box-whisker plots and scatter plots for showing the quantitative performance distributions of all concepts in Case Study II (Yang et al., 2020)

TOC

6.4.1.3 Multi-objective optimization model re-formulation

The initial MOO model is re-formulated based on the acquired knowledge. The specific re-formulation actions include quantitative objective reduction, qualitative constraint addition, and divergent concept generation, as suggested by the acquired knowledge.

Specifically, the quantitative objectives other than Min_EUI, Min_OLA, and Max_AU are removed; a qualitative constraint Cons_Aesthetics is taken into account; for generating Concept 1_1, ExtrusionHeight and ExtraOpeningWidth are added and WindowWidth is revised; for generating Concept 2_1, WindowWidthNS is revised; for generating Concept 3_1, BuildingOrientation is added and WindowWidth is revised. The blue texts in TABLE 6.1 (4-6) represent the newly added and revised variables in this re-formulation iteration.

6.4.2 The second-time re-formulation

6.4.2.1 Data generation

The second group of 300 data sets is generated for analysis. It is derived based on an updated automation workflow. The workflow is updated by uploading the first re-formulated MOO model.

6.4.2.2 Information and knowledge extraction

Based on the second group of 300 data sets, a category of knowledge is extracted, namely knowledge about promising new concepts besides the existing ones, as exemplified in FIG.3.5.

Knowledge about promising new concepts besides the existing ones

Finding quantitatively promising new concepts in the second re-formulation iteration is similar to that in the first re-formulation iteration. As a result, the quantitative performance distributions of Concept 1_1, Concept 2_1, and Concept 3_1, are visualized in box-whisker plots and scatter plots (see FIG.6.8). Based on the result, distribution information is extracted and interpreted, so as to acquire desired knowledge, as described below.

1 Regarding Concept 1_1, AU is improved as expected, but unexpectedly, OLA becomes worse. This may indicate that daylight reflected by the north inclined opaque element is too concentrated in some spots. A possible strategy to fix this issue can be reflecting or redirecting daylight into a broader range. Concept 1_2 is created based on this strategy - by making the north opaque elements vertical and adding shading elements.

2 Regarding Concept 2_1, EUI is improved, OLA and AU are maintained, as expected. In order to obtain an even better EUI, a more aggressive strategy can be increasing daylight from the north. Concept 2_2 is created based on this strategy – by expanding the openings of the roof bottom surface and enlarging the size of the north-facing windows.

3 Regarding Concept 3_1, EUI is improved as expected, but unexpectedly, AU does not become better. This may indicate that the sawtooth geometry has difficulties spreading daylight evenly over the space. A possible strategy to fix this issue can be reflecting or redirecting daylight into a broader range. Concept 3_2 is created based on this strategy – by expanding the openings of the roof bottom surface and adding shading elements.

Finding qualitatively promising new concepts in the second re-formulation iteration is similar to that in the first re-formulation iteration, which requires human subjectivity. Here, it is assumed that Concept 1_2 and Concept 2_2 have less promising Aesthetics performance, Concept 3_2 has more promising Aesthetics performance, and that no other new concepts are generated only for an aesthetical purpose.

Overall, in this stage, the new Concept 1_2, Concept 2_2, and Concept 3_2 are considered helpful for improving quantitative performances and maintaining acceptable qualitative performance.

6.4.2.3 Multi-objective optimization model re-formulation

The latest MOO model is further re-formulated based on the acquired knowledge. The specific re-formulation actions include divergent concept generation, as suggested by the acquired knowledge.

Specifically, for generating Concept 1_2, ShadingAngleA is added; for generating Concept 2_2, ExtrusionHeight and ExtraOpeningWidth are added, OpeningWidth is revised, and WindowWidthNS is removed; for generating Concept 3_2, ShadingAngleB is added and OpeningWidth is revised. The red texts in TABLE 6.1 (7-9) represent the newly added and revised variables in this re-formulation iteration.

6.4.3 The third-time re-formulation

6.4.3.1 Data generation

The third group of 300 data sets is generated for analysis. It is derived based on an updated automation workflow. The workflow is updated by uploading the second re-formulated MOO model.

6.4.3.2 Information and knowledge extraction

Based on the third group of 300 data sets, a category of knowledge is extracted, namely knowledge about promising existing concepts among all explored ones, as exemplified in FIG.3.5.

Knowledge about promising existing concepts among all explored ones

Finding quantitatively promising existing concepts in the third re-formulation iteration, is similar to that in the first re-formulation iteration. The quantitative data to be analyzed here includes all 900 data sets (i.e., all three groups of data sets). As a result, ten clusters of samples are generated and visualized in a parallel coordinate chart (see FIG.6.9, left). Based on the result, clustering information is extracted and interpreted, so as to acquire desired knowledge, as described below.

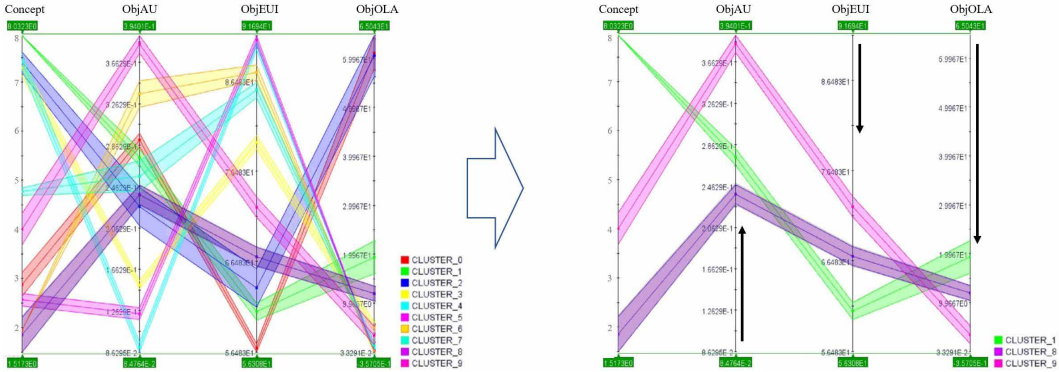


FIG. 6.9 Clusters in the third re-formulation iteration (Yang et al., 2020)

Note: 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.

1. Samples in CLUSTER_1 belonging to Concept 2_2, samples in CLUSTER_8 belonging to Concept 1_0 and Concept 1_1, and samples in CLUSTER_9 belonging to Concept 2_0 and Concept 2_1 can reach the chosen objectives simultaneously (see FIG.6.9, right).

2. Concept 1_0, Concept 1_1, Concept 2_0, Concept 2_1, and Concept 2_2 can be considered as quantitatively more promising existing concepts, given that the identified clusters of samples belong to these five concepts.

3. Concept 3_0, Concept 3_1, Concept 3_2, and Concept 1_2 can be considered as quantitatively less promising existing concepts, given that the identified clusters of samples do not belong to these four concepts.

It is worth noting that concepts generated in later re-formulation iterations may not necessarily outperform those generated in earlier re-formulation iterations due to possible inaccuracy of prior knowledge. 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.6.8. That is, the inaccuracy of prior knowledge can be corrected if it occurs during the re-formulation.

Finding qualitatively promising existing concepts in the third re-formulation iteration is similar to that in the first re-formulation iteration, which requires human subjectivity. Here, it is assumed that Concept 1_0, Concept 1_1, Concept 1_2, Concept 2_0, Concept 2_1, and Concept 2_2 have less promising Aesthetics performance, and Concept 3_0, Concept 3_1, and Concept 3_2 have more promising Aesthetics performance.

Overall, in this stage, the existing Concept 1_0, Concept 1_1, Concept 2_0, Concept 2_1, and Concept 2_2 are considered quantitatively more promising; the existing Concept 3_0, Concept 3_1, and Concept 3_2 are considered qualitatively more promising; the existing Concept 1_2 is considered quantitatively and qualitatively less promising.

6.4.3.3 Multi-objective optimization model finalization

The latest MOO model is further re-formulated based on the acquired knowledge. The specific re-formulation actions include convergent concept selection, as suggested by the acquired knowledge.

For different purposes of study, different MOO models can be created. In this study, MOO models 0-9 are created by selecting different sets of design variables, as described below.

The first purpose is to study whether the Subtype-II method (i.e., dynamic method) is better than the traditional method (defined in FIG.3.1). For this, MOO model 0 is created based on the traditional method. It means that all the initial design variables are kept unchanged to create the model. Moreover, MOO models 1, 2, 4, 5, and 8 are created based on the acquired knowledge. First, the way of selecting concepts is assumed to be prioritizing quantitative performances; then, the variables related to the quantitatively more promising concepts are selected (i.e., Concept 1_0, Concept 1_1, Concept 2_0, Concept 2_1, and Concept 2_2). Given that these selected concepts can also be denoted by using the "*Concept*" values 1, 2, 4, 5, and 8, the resulting MOO models are denoted by using the same numbers.

The second purpose is to study how the choice of the high-level variable values (i.e., "*Concept*" values) may affect the optimization results. For this, MOO models 3, 6, and 9 are created. First, the way of selecting concepts is assumed to be prioritizing qualitative performances; then, the variables related to the qualitatively more promising concepts are selected (i.e., Concept 3_0, Concept 3_1, and Concept 3_2). Moreover, MOO model 7 is created. First, the way of selecting concepts is assumed to be prioritizing neither quantitative performances nor qualitative performances; then, the variables related to the quantitatively and qualitatively less promising concept are selected (i.e., Concept 1_2). Given that the selected concepts can be also denoted by using the "*Concept*" values 3, 6, 9 and 7, the resulting MOO models are denoted by using the same numbers.

The fourth purpose is to study how the use of a purely random initial population may affect the optimization results. For this, MOO models 1-9 are used.

6.4.4 Output of Phase-II

The main output of Phase-II includes a final MOO model which includes a final set of performance objectives, constraints, and design variables (i.e., MOO model 1). Moreover, it can also include other MOO models for comparison purposes. In this case, all of them are to be used in the next phase.

6.5 Phase-III utilizing a directed initial population

Phase-III (i.e., Optimization Problem Solving) of the Subtype-II method (i.e., dynamic method) utilizes a directed initial population.

In Case Study II, the directed initial population consists of samples selected from high-performing clusters. Such clusters can be found from the parallel coordinate chart shown in FIG.6.9. In order to study relevant hypotheses, different initial populations (i.e., directed and purely random initial populations) are used in combination with different MOO models. They are used to set up multiple MOO runs which are then executed (Section 5.6.1). MOO results are compared (Section 5.6.2), in order to extract knowledge about the hypotheses as the main output of this phase (Section 5.6.3).

6.5.1 Multi-objective optimization setup

6.5.1.1 Setup for studying hypothesis one

It is hypothesized that adopting the Subtype-II method (i.e., dynamic method) can help achieve a quantitatively and qualitatively better Pareto front, compared with adopting the traditional method (defined in FIG.3.1). For studying this hypothesis, two groups of MOO runs are set up (i.e., MOO run A and B), as described below.

For studying the effects of adopting the traditional method, MOO run A is set up. It is based on: MOO model 0 (in which the "*Concept*" is treated as a high-level variable with the range of 1, 2, and 3) and a purely random initial population (selected from the entire design space). It includes one MOO run.

For studying the effects of adopting the Subtype-II method (i.e., dynamic method), MOO run B is set up. It is based on: MOO model 1, 2, 4, 5, and 8 (in each of which the "*Concept*" is treated as a constant value, namely 1, 2, 4, 5, and 8 respectively) and directed initial populations (selected from high-performing clusters having the chosen "*Concept*" values). It includes five MOO runs.

6.5.1.2 Setup for studying hypothesis two

It is hypothesized that factors including the choice of high-level variable values and initial populations may affect the behaviors of Subtype-II method (i.e., dynamic method). More specifically, factors of choosing different high-level variable values (i.e., "*Concept*" values) and utilizing a purely random initial population may affect final MOO results. For studying this hypothesis, three more groups of MOO runs are set up (i.e., MOO runs C to E), as described below.

For studying the impacts of choosing different high-level variable values (i.e., "*Concept*" values), MOO runs C and D are set up. MOO run C is based on: MOO models 3, 6, and 9 (in each of which the "*Concept*" is treated as a constant value, namely 3, 6, and 9 respectively) and directed initial populations (selected from high-performing clusters having the chosen "*Concept*" values). It includes three MOO runs. MOO run D is based on: MOO model 7 (in which the "*Concept*" is treated as a constant value, namely 7) and a directed initial population (selected from high-performing clusters having the chosen "*Concept*" value). It includes one MOO run.

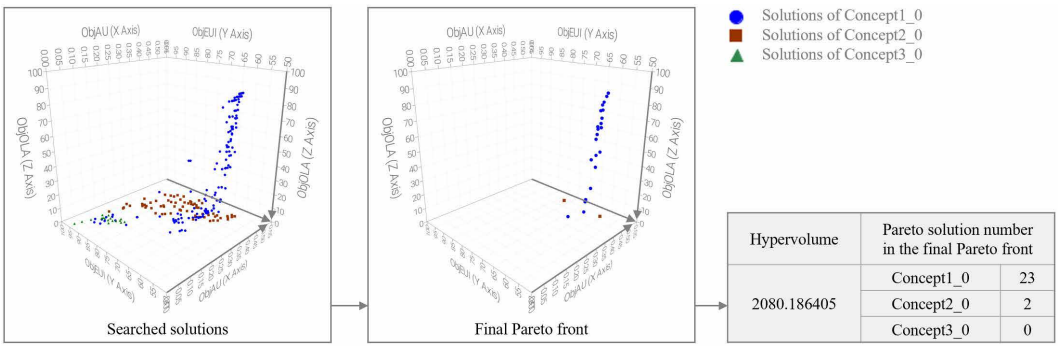
For studying the impacts of utilizing a purely random initial population, MOO run E is set up. It is based on: MOO models 1-9 and random initial populations (selected from the entire design space). It includes nine MOO runs.

6.5.1.3 Optimization execution

There is a total of nineteen optimizations to execute. For all these optimizations, the same MOO algorithm and settings are used: Non-dominant Sorting Genetic Algorithm II (NSGA-II), a population size of 30, and 10 generations etc. All these optimizations are executed on a 6-Core (12-Thread) computer. The total time for executing each optimization is within 1 day; and the average time for evaluating each solution is around 2.5-4.5 minutes.

6.5.2 Multi-objective optimization result comparison

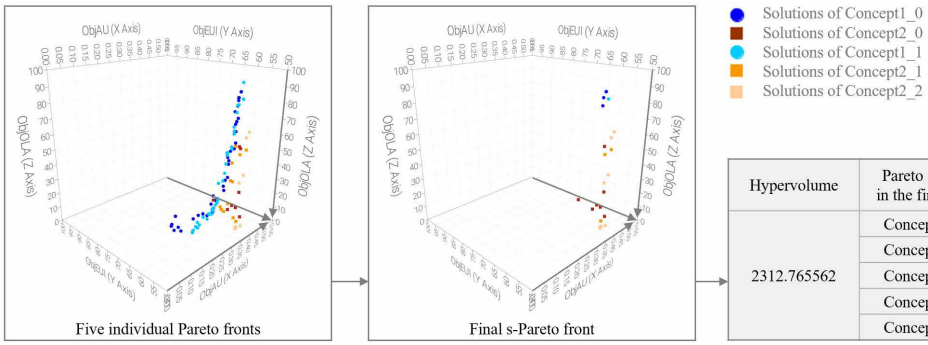
The MOO results include the (quantitative and qualitative) data of Pareto solutions. The data is organized in ways that facilitate the comparison of Pareto solutions (see FIG.6.10 - FIG.6.14). For instance, the performance values of Pareto solutions are plotted in the same 3D space; the numbers of Pareto solutions belonging to different concepts and the hypervolume values of Pareto fronts are presented. Moreover, the optimization result comparison is summarized in TABLE 6.3.



Hypervolume	Pareto solution number in the final Pareto front	
2080.186405	Concept1_0	23
	Concept2_0	2
	Concept3_0	0

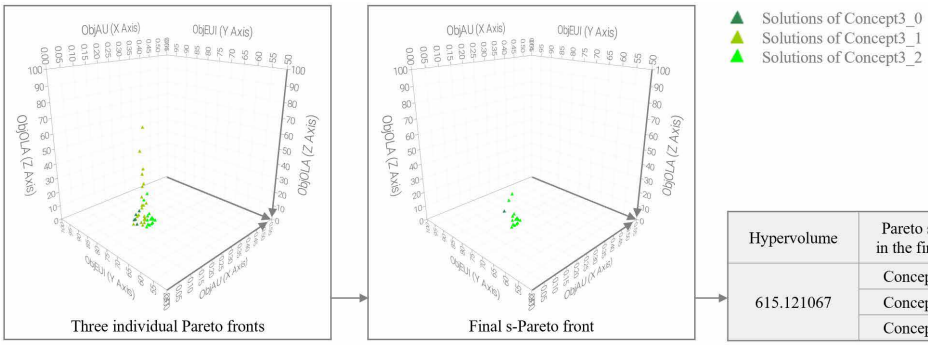
FIG. 6.10 Optimization results derived from MOO run A in Case Study II

TOC



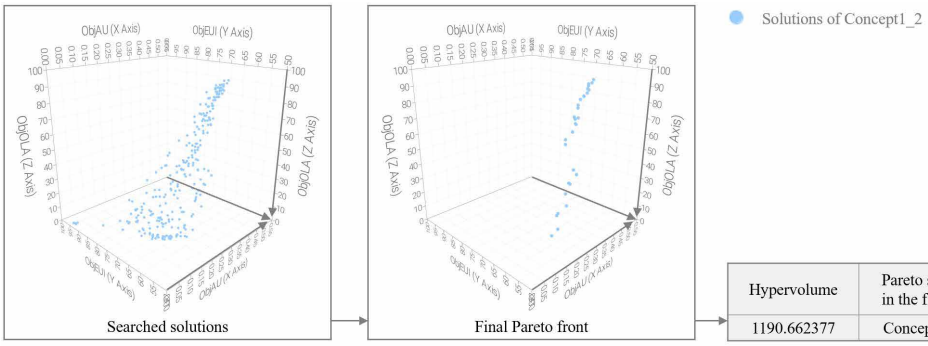
Hypervolume	Pareto solution number in the final s-Pareto front	
2312.765562	Concept1_0	4
	Concept2_0	7
	Concept1_1	1
	Concept2_1	3
	Concept2_2	8

FIG. 6.11 Optimization results derived from MOO run B in Case Study II



Hypervolume	Pareto solution number in the final s-Pareto front	
615.121067	Concept3_0	1
	Concept3_1	0
	Concept3_2	17

FIG. 6.12 Optimization results derived from MOO run C in Case Study II



Hypervolume	Pareto solution number in the final Pareto front	
1190.662377	Concept1_2	31

FIG. 6.13 Optimization results derived from MOO run D in Case Study II

TOC

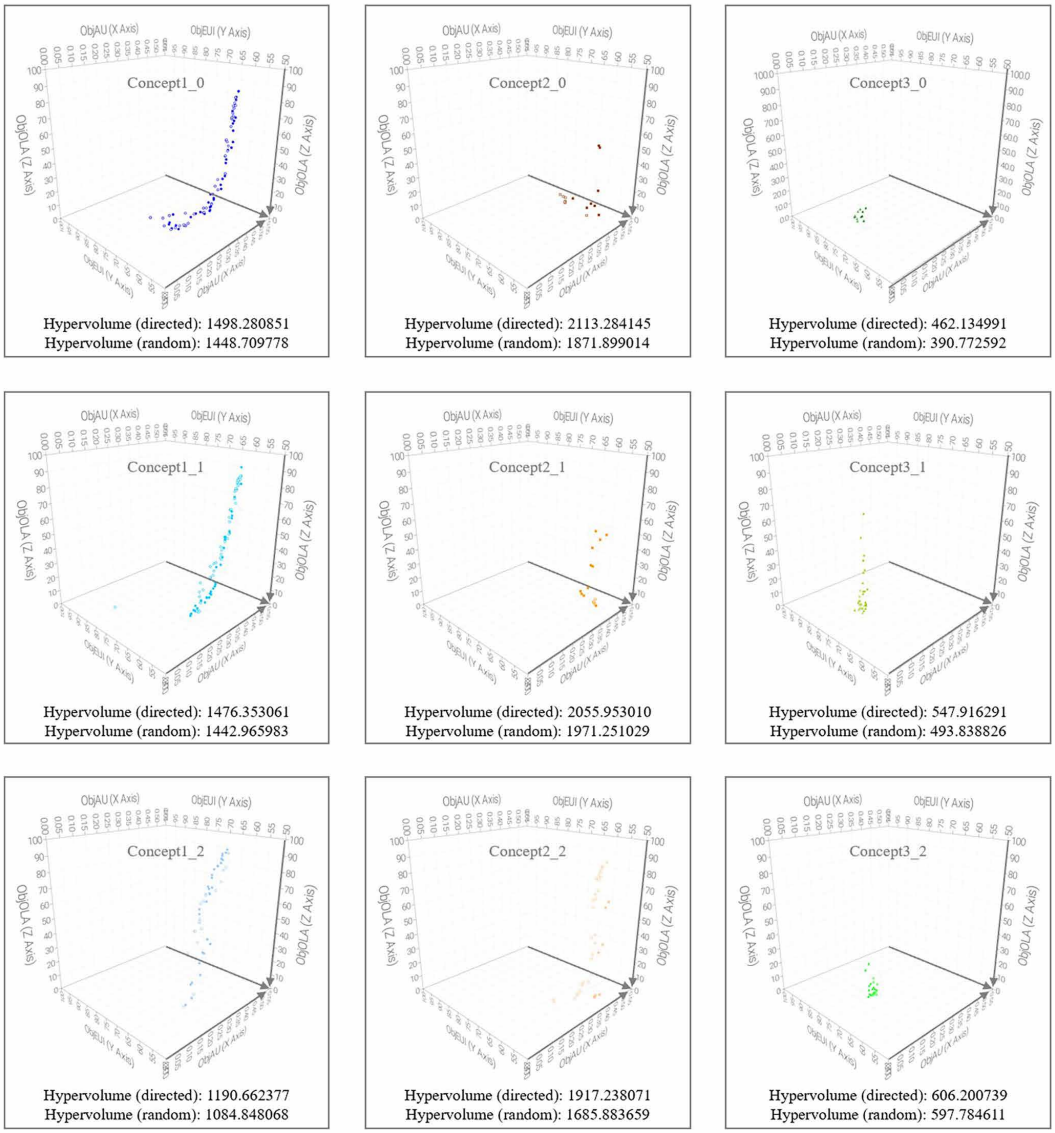


FIG. 6.14 Optimization results derived from MOO run E in Case Study II

TOC

TABLE 6.3 Summary of optimization result comparison in Case Study II

Purposes of the verification	Optimization result comparison	Quantity	Pertinence	Proximity and diversity	Geometric preference compliance	Geometric variation appropriateness
Verify the benefits of adopting the Subtype-II method	s-Pareto solutions from MOO run B (compared with Pareto solutions from MOO run A)	-	x	+	x	+
Verify the impacts of choosing different high-level variable values (i.e., "Concept" values)	s-Pareto solutions from MOO run C (compared with s-Pareto solutions from MOO run B)	-	-	-	+	-
	Pareto solutions from MOO run D (compared with s-Pareto solutions from MOO run B)	+	+	-	x	-
Verify the impacts of utilizing a purely random initial population	Pareto solutions from MOO run E (compared with Pareto solutions from MOO run B, C, D)	x	x	-	x	x

Note: "+" represents better results or positive impacts; "-" represents worse results or negative impacts; "x" represents similar results or no significant impacts.

Some issues are important when comparing the MOO results. First, it is important to know the concept of an s-Pareto front (Mattson and Messac, 2003; Mattson and Messac, 2005). The s-Pareto front is actually a new kind of Pareto front derived by combining multiple individual Pareto fronts. This combination applies the Pareto dominance principle to all the solutions on the individual Pareto fronts, thus figuring out a new set of non-dominated solutions that constitute the s-Pareto front. Moreover, it is also important to make clear what an ideal Pareto front (or s-Pareto front) is in the context that highlights sparking new design possibilities. As explained in Section 3.2.3.3, an ideal Pareto front (or s-Pareto front) in this context should have good proximity and diversity, good pertinence, good geometric preference compliance, and good geometric variation appropriateness. Here, good pertinence means that Pareto solutions are within a large region of interest; good geometric variation appropriateness means that Pareto solutions have a high degree of geometric variations.

6.5.2.1 Result comparison for verifying hypothesis one

To verify the benefits of adopting the Subtype-II method (i.e., dynamic method), the s-Pareto solutions from MOO run B (see FIG.6.11) are compared with the Pareto solutions from MOO run A (see FIG.6.10).

The s-Pareto solutions from MOO run B (1) have a quantity less similar to the size of the initial population; (2) are within a similar region of interest; (3) lead to a higher hypervolume value 2312.77; (4) are similarly compliant with the geometric preference; and (5) have a higher degree of geometric variations. These facts confirm that adopting the Subtype-II method (i.e., dynamic method) can help achieve a better Pareto front in terms of the proximity and diversity, and geometric variation appropriateness, compared with adopting the traditional method.

6.5.2.2 Result comparison for verifying hypothesis two

To verify the impacts of choosing different high-level variable values (i.e., "*Concept*" values), the s-Pareto solutions from MOO run C (see FIG.6.12) and the Pareto solutions from MOO run D (see FIG.6.13) are respectively compared with the s-Pareto solutions from MOO run B (see FIG.6.11).

First, the s-Pareto solutions from MOO run C (1) have a quantity less similar to the size of the initial population; (2) are within a smaller region of interest; (3) lead to a lower hypervolume value 615.12; (4) are more compliant with the geometric preference; and (5) have a lower degree of geometric variations. These facts confirm that choosing "*Concept*" values 3, 6, and 9 can have negative impacts on the quantity, pertinence, proximity and diversity, and geometric variation appropriateness of the s-Pareto front, but have positive impacts on the geometric preference compliance.

Second, the Pareto solutions from MOO run D (1) have a quantity more similar to the size of the initial population; (2) are within a larger region of interest; (3) lead to a lower hypervolume value 1190.66; (4) are similarly compliant with the geometric preference; and (5) have a lower degree of geometric variations. These facts confirm that choosing "*Concept*" values 7 can have negative impacts on the proximity and diversity and geometric variation appropriateness of the Pareto front but have positive impacts on the quantity and pertinence.

To verify the impacts of utilizing a purely random initial population, the nine sets of Pareto solutions from MOO run E (see FIG.6.14) are compared with the corresponding nine sets of Pareto solutions from MOO runs B, C, and D (see FIG.6.11 – FIG.6.13). The Pareto solutions derived by utilizing a purely random initial population are roughly similar to those derived by utilizing a directed initial population, in terms of the quantity, pertinence, geometric preference compliance, and geometric variation appropriateness; but the former Pareto solutions have

lower hypervolume values. These facts confirm that utilizing a purely random initial population can have negative impacts on the proximity and diversity of the Pareto fronts.

6.5.3 Output of Phase-III

The main output of Phase-III includes not only final Pareto solutions and s-Pareto solutions, but also various knowledge derived by comparing these solutions. In this case, the knowledge obtained in this phase is about the benefits of adopting the Subtype-II method (i.e., dynamic method) and the factors affecting the behaviors of the method.

6.6 Discussion

In this case study, the benefits of adopting the Subtype-II method (i.e., dynamic method) are derived from the incorporation of the cyclical knowledge-supported re-formulation process (Section 6.4). During the knowledge extraction process, divergent knowledge is obtained (i.e., the knowledge for generating new design solutions in the design space). With the help of this knowledge, new promising design variables are added to enrich concepts divergently, thus facilitating the achievement of a more reliable optimization model and more reliable optimal solutions.

Human designers play an influential role during the cyclical re-formulation process. Their influence has two sides. On one side, they have the advantages to support qualitative thinking and divergent thinking; while on the other side, they may cause negative impacts on final optimal results when using the Subtype-II method (i.e., dynamic method) in different ways. For instance, when designers decide to choose high-level variable values (i.e., "*Concept*" values) other than those that are considered quantitatively promising, negative impacts on the final Pareto front can occur; when designers decide to utilize a purely random initial population (rather than a directed initial population), the starting point for optimization is relatively low, which can also lead to negative impacts on the final Pareto front.

6.7 Conclusion

This chapter concludes by summarizing the main research results (Section 6.6.1); identifying possible applications of the Subtype-II method (i.e., dynamic method) (Section 6.6.2); and providing concluding remarks (Section 6.6.3).

6.7.1 Main research results

The main research results of this chapter include the following:

- The overall process of the Subtype-II method (i.e., dynamic method) has been demonstrate in Case Study II. This subtype method includes a three-time Optimization Problem Re-Formulation (Re-OPF) process that focuses on adding new variables (i.e., enriching more concepts divergently).

- The benefits of adopting the Subtype-II method (i.e., dynamic method) have been verified. As confirmed by the optimization result comparison, adopting this subtype method can help achieve a better s-Pareto front in terms of proximity and diversity, and geometric variation appropriateness, compared with adopting the traditional method.

- The factors affecting the behaviors of the Subtype-II method (i.e., dynamic method) have been verified. As confirmed by the optimization result comparison, factors of choosing different high-level variable values (i.e., “Concept” values), and utilizing a purely random initial population can have varying impacts on the optimization results, including positive and negative impacts, as summarized in TABLE 6.3. Most of the impacts are negative and need to be avoided; but it is also worth noting the positive impacts, namely the impacts of choosing different high-level variable values (i.e., “Concept” values) on the quantity, pertinence, and geometric preference compliance.

6.7.2 Possible applications

The Subtype-II method (i.e., dynamic method) can be further applied at least in the following two scenarios:

- First, this subtype method can be applied in a scenario that involves more complete performance measures and more complex building geometries. In Case Study II, the method has just been used to deal with the hypothetical project that involves incomplete performance measures and simplified building geometries (for the convenience of demonstrating the method). However, for a practical project, it is often necessary to consider more complete performance measures and more complex building geometries.

- Second, this subtype method can be applied in a scenario that involves multiple re-formulations of performance measures. In Case Study II, the method has just been used to deal with the hypothetical project that involves one re-formulation of performance measures in the Optimization Problem Initial-Formulation (Initial-OPF) phase (for the convenience of focusing on design variable re-formulation). However, for a practical project, it can also be necessary to consider the iterative re-formulation of various performance measures.

6.7.3 Concluding remarks

In conclusion, the Subtype-II method (i.e., dynamic method) is applicable in the conceptual design of a sports training hall with the aid of the new Gh-mF node. It is particularly useful for the relatively early sub-phase of the conceptual design where the main purpose is to spark new design possibilities. Thanks to the three-time Optimization Problem Re-Formulation (Re-OPF) process, this subtype method can help achieve a better s-Pareto front in terms of the proximity and diversity and geometric variation appropriateness, compared with adopting the traditional method (defined in FIG.3.1). These benefits can be affected by factors like choosing different high-level variable values (i.e., "*Concept*" values), and utilizing a purely random initial population.

References

Al-Obaidi, K. M., & Rahman, A. M. A. (2016). Top lighting systems for improving the indoor environment: a review. In: M. Ahmad, M. Ismail, & S. Riffat (Eds.), Renewable Energy and Sustainable Technologies for Building and Environmental Applications (pp. 117-136). Springer, Cham. ISBN: 978-3-319-31838-7.

Beltran, L. O. (2005). Top lighting for the Tropics. In: D. K. Raydan, H. K. Melki (Eds.), Environmental Sustainability: The Challenge of Awareness in Developing Societies (pp. 297-300). NDU Press, Lebanon. ISBN: 9953418691.

CIBSE (1999). Daylighting and Window Design. The Chartered Institute of Building Service Engineers (CIBSE), London. ISBN: 0900953985.

Harntaweegonsa, S., & Beltran, L. (2007). Daylighting Performance of Toplighting Systems in the Hot and Humid Climate of Thailand. In: Solar 2006: Renewable Energy - Key to Climate Recovery (pp. 613-618). Curran Associates, New York. ISBN: 978-1604232882.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323. DOI: https://doi.org/10.1145/331499.331504.

Kohonen, T. (2001). Self-Organizing Maps. Third edition. Springer, Berlin, Heidelberg. ISBN: 978-3-540-67921-9.

Lechner, N. (2014). Heating, Cooling, Lighting: Sustainable Design Methods for Architects. Fourth Edition. John Wiley & Sons, Hoboken. ISBN: 978-1-118-58242-8.

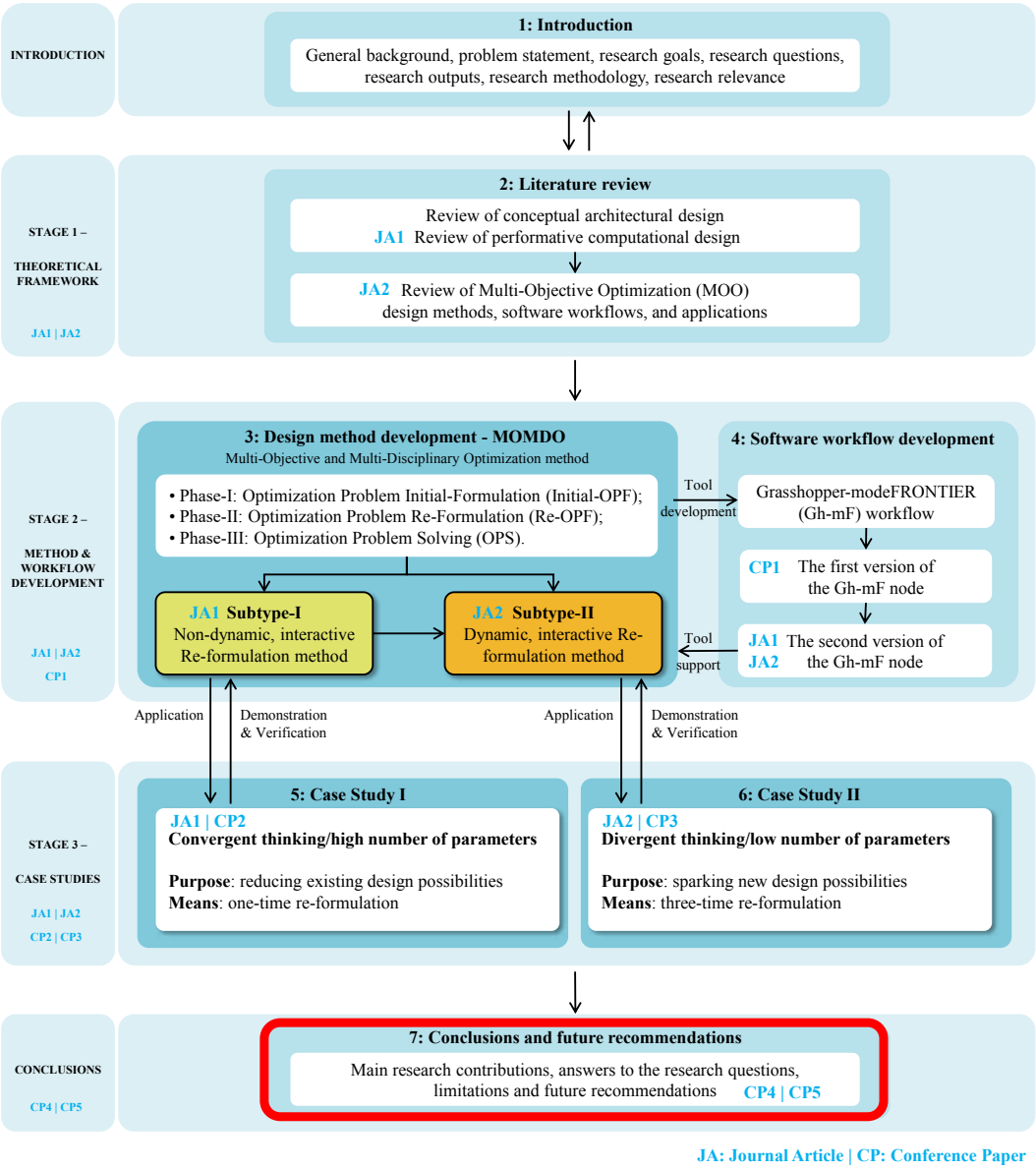
Mattson, C. A., & Messac, A. (2003). Concept selection using s-Pareto frontiers. AIAA journal, 41(6), 1190-1198. DOI: https://doi.org/10.2514/2.2063.

Mattson, C. A., & Messac, A. (2005). Pareto frontier based concept selection under uncertainty, with visualization. Optimization and Engineering, 6(1), 85-115. DOI: https://doi.org/10.1023/B:OPTE.0000048538.35456.45.

Mavridou, T., & Doulos, L. T. (2019). Evaluation of different roof types concerning daylight in industrial buildings during the initial design phase: Methodology and case study. Buildings, 9(7), 170. DOI: https://doi.org/10.3390/buildings9070170.

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, 103251. DOI: https://doi.org/10.1016/j.autcon.2020.103251.

Yang, D., Sun, Y., di Stefano, D., & Turrin, M. (2017). A computational design exploration platform supporting the formulation of design concepts. In: M. Turrin, B. Peters, W. O'Brien, R. Stouffs, & T. Dogan (Eds.), 2017 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 35-42). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-88878-6.


INTRODUCTION
1: Introduction
General background, problem statement, research goals, research questions, research outputs, research methodology, research relevance
STAGE 1 – THEORETICAL FRAMEWORK
JA1 | JA2
2: Literature review
Review of conceptual architectural design
JA1 Review of performative computational design
JA2 Review of Multi-Objective Optimization (MOO) design methods, software workflows, and applications
STAGE 2 – METHOD & WORKFLOW DEVELOPMENT
JA1 | JA2
CP1
3: Design method development - MOMDO
Multi-Objective and Multi-Disciplinary Optimization method
• Phase-I: Optimization Problem Initial-Formulation (Initial-OPF);
• Phase-II: Optimization Problem Re-Formulation (Re-OPF);
• Phase-III: Optimization Problem Solving (OPS).
JA1 Subtype-I
Non-dynamic, interactive Re-formulation method
JA2 Subtype-II
Dynamic, interactive Re-formulation method
Tool development
4: Software workflow development
Grasshopper-modeFRONTIER (Gh-mF) workflow
CP1 The first version of the Gh-mF node
JA1 JA2 The second version of the Gh-mF node
Tool support
Application
Demonstration & Verification
Application
Demonstration & Verification
STAGE 3 – CASE STUDIES
JA1 | JA2
CP2 | CP3
5: Case Study I
JA1 | CP2
Convergent thinking/high number of parameters
Purpose: reducing existing design possibilities
Means: one-time re-formulation
6: Case Study II
JA2 | CP3
Divergent thinking/low number of parameters
Purpose: sparking new design possibilities
Means: three-time re-formulation
CONCLUSIONS
CP4 | CP5
7: Conclusions and future recommendations
Main research contributions, answers to the research questions, limitations and future recommendations CP4 | CP5
JA: Journal Article | CP: Conference Paper

TOC

7 Conclusions and future recommendations

This chapter concludes the research. First, it summarizes the main contributions of this research (Section 7.1). Then, it presents comprehensive answers to all research questions to provide a more inclusive response to the contributions of this research (Section 7.2). Last, it points out the research limitations and future recommendations (Section 7.3).

7.1 Main research contributions

In current optimal-design paradigms, there are limitations related to supporting ill-structured optimization problems. Specifically, there is often a lack of a way to ensure the achievement of a reliable optimization problem. Addressing this limitation is important for obtaining reliable design solutions. It is particularly true in the conceptual design of complex buildings like indoor sports halls, where many multi-disciplinary performance requirements and multi-scale design concepts can be involved but are usually ill-defined.

The main contributions of this research relate to a new optimal-design method proposed to deal with ill-defined conceptual architectural design. They can be summarized as follows:

- Proposing a novel Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method featured with dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) for achieving more reliable high-performing solutions in conceptual architectural design.

- Providing a flexible or an open optimal-design system that allows for considering quantitative and qualitative performance measures from different disciplines, and for considering divergent and convergent design variables from different scales.

- Providing a human-computer interactive knowledge extraction process that can help designers make more informed, performance-based, early decisions on the choice of performance measures and design variables, and thus drive a conceptual design process forward towards promising directions.

- Substantiating the relevance of Optimization Problem Re-Formulation (Re-OPF) by two conceptual indoor sports hall design cases that concern, respectively, refining an existing concept convergently and enriching new concepts divergently.

In fact, all the above contributions specifically relate to **the knowledge-supported, dynamic and interactive Optimization Problem Re-Formulation (Re-OPF)** in the proposed method. Therefore, such re-formulation is a key innovation which differentiates the proposed method from other methods in the field of architectural design optimization.

7.2 Answers to the research questions

By answering the main research question and the sub-questions, a more in-depth understanding of the research contributions can be achieved.

Main question: how to assist architects and engineers to extract useful information and knowledge to support dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) during ill-defined conceptual design?

The main question relates to the main goal of developing a Multi-Objective and Multi-Disciplinary Optimization method suitable for use in ill-defined conceptual architectural design, by leveraging information and knowledge extraction to support dynamic and interactive Optimization Problem Re-Formulation (Re-OPF).

Information and knowledge extraction is the core of Optimization Problem Re-Formulation (Re-OPF). It relies on human-computer interaction. During the extraction, relevant computational techniques, especially advanced quantitative

data analysis techniques, are used to covert design and performance data into quantitative information; meanwhile, human designers can extract qualitative information based on their subjective judgments, and then interpret the quantitative and qualitative information into useful knowledge. Based on the knowledge, optimization problems are re-formulated in a more informed manner, thus driving the design process forward.

Regarding the knowledge brought by this research, it includes not just disciplinary knowledge in a given domain or for a particular case, but more importantly the knowledge (i.e., the way) of structuring or re-structuring disciplinary knowledge. The latter knowledge seems more important, given that contemporary buildings are getting increasingly complex in terms of form and performance. Not only can it help novice designers supplement their missing disciplinary knowledge, but it can also inspire expert designers to apply their already known knowledge in more creative ways.

1 **How to improve the reliability of an design task and an optimization problem, and to what extent do current optimal-design methods deal with this issue?**

This sub-question (discussed in Chapter 2) relates to the sub-goal of ascertaining a way to achieving a reliable design task and a reliable optimization problem and of identifying the general state of optimal-design methods in supporting this way.

Dynamic and interactive design task re-definition is a potential means to achieve a more reliable design task. For an ill-defined design task, the priority is task framing (more precisely task re-framing) rather than task-solving. The task re-framing can reduce inaccuracies or uncertainties in a design task and is good for obtaining more reliable design solutions. The task-framing can be aided by the use of optimization (i.e., partially converting an ill-defined design task into an ill-structured optimization problem).

Dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) is a potential means to achieve a more reliable optimization problem. For an ill-structured optimization problem, problem-framing (more precisely problem re-framing) is more essential than problem-solving. An ideal optimal-design paradigm should shift the priority from Optimization Problem Solving (OPS) to Optimization Problem Re-Formulation (Re-OPF). Although inaccuracies or uncertainties are often unavoidable when converting a design task into an optimization problem, it is worth reducing them to the fullest possible extent.

The dynamic and interactive features are important for the above re-definition and re-formulation. The former allows continuous knowledge extraction, while the latter supports both quantitative and qualitative considerations and both divergent and convergent considerations. These features make it possible to re-consider an ill-defined design task and an ill-structured optimization problem in a more informed and inclusive manner, thus helping to achieve a less ill-defined design task and a less ill-structured optimization problem.

According to a preliminary review, current optimal-design methods rarely consider Optimization Problem Re-Formulation (Re-OPF), not to mention dynamic and interactive re-formulation. This leads to further reviews that focus on Multi-Objective Optimization (MOO).

2 **To what extent: (1) is dynamic and interactive Optimization Problem Re-Formulation (Re-OPF) supported? (2) are necessary computational techniques provided? (3) are optimal-design methods applied to the conceptual design of sports buildings?**

This sub-question (discussed in Chapter 2) relates to the sub-goal of identifying the state of the art of Multi-Objective Optimization (MOO) design methods, software workflows, and application to the conceptual design of sports buildings.

A detailed review of Multi-Objective Optimization (MOO) design methods has shown that existing methods may adopt different computational techniques (e.g., parametric geometric modeling, sampling algorithms, multi-disciplinary simulation modeling, Multi-Objective Optimization algorithms, quantitative data analysis, and qualitative data visualization) in various ways, but these techniques are not necessarily used in a way that facilitates the Optimization Problem Re-Formulation (Re-OPF). More specifically, there is a lack of Multi-Objective Optimization (MOO) design methods that have incorporated dynamic and interactive Optimization Problem Re-Formulation (Re-OPF); and the few methods that have done so still have room for improvement, especially in terms of information and knowledge extraction.

A broad review of Multi-Objective Optimization (MOO) software workflows has indicated that Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software have great potential to offer the aforementioned computational techniques, but these two types of software are not usually used together. In other words, there is a lack of Multi-Objective Optimization (MOO) software workflows that have integrated Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software; and a software workflow that have done so still has limitations, especially in terms of software integration (i.e., the integration of Grasshopper and modeFRONTIER).

Another review has been conducted to understand the state of the art in applying optimal-design methods to the conceptual design of sports buildings. According to the review, optimal-design methods, especially Multi-Objective Optimization (MOO) design methods, are not commonly used during the conceptual design of sports buildings; and the conceptual sports building design processes that utilize optimal-design methods have not fully considered building geometry interaction and building performance conflicts.

3 **How to arrange actions and adopt necessary computational techniques for the proposed optimal-design method?**

This sub-question (discussed in Chapter 3) relates to the sub-goal of developing an optimal-design method that enables dynamic and interactive Optimization Problem Re-Formulation (Re-OPF). It is directly associated with the main question.

The proposed method consists of three phases. In these phases, several groups of general actions are arranged and several types of computational techniques are adopted in flexible ways.

- **Phase-I:** Optimization Problem Initial-Formulation (Initial-OPF)

This phase deals with the formulation of an initial MOO model. It involves two groups of general actions.

Initial concept generation: designers can brainstorm one or multiple initial design concepts based on their own knowledge or past experiences, and then choose desired design variables to fulfill the initial performance requirements that they consider are important.

Initial MOO model formulation: based on the selected initial design variables and performance measures, designers can create their own initial parametric geometry model and integrate it with multiple simulation models by using flexible parametric geometric modeling and multi-disciplinary simulation modeling techniques (e.g., two-level variable structure, modular programming). As a result, an initial MOO model consisting of an initial set of performance objectives, constraints and design variables is achieved.

- **Phase-II:** Optimization Problem Re-Formulation (Re-OPF)

  This phase deals with the re-formulation of the initial (or latest) MOO models. It can iterate through the following three groups of general actions for one or multiple times.

  Data generation: samples can be selected automatically by using an advanced sampling algorithm (e.g., Uniform Latin Hypercube sampling) from the design space defined by the initial (or latest) MOO model. The chosen samples, as a representation of the entire design space, are used to generate qualitative data sets (i.e., images showing building geometries) and quantitative data sets (i.e., input values defining building geometries and output values representing performance results) automatically in sequential order.

  Information and knowledge extraction: to extract information about quantitative performances, the quantitative data sets can be analyzed by computers using advanced quantitative data analysis techniques (e.g., Self-Organizing Map, Hierarchical Clustering, Smoothing Spline Analysis of Variance); to extract information about qualitative performances, the qualitative data sets can be observed by humans with the aid of useful qualitative data visualization techniques (e.g., combined data visualization). Then, designers can interpret these two types of information and synthesize them into new knowledge about which design variables and performance measures should be added or removed.

  MOO model re-formulation: based on the extracted knowledge, designers can re-define and integrate the initial (or latest) parametric geometry model and simulation models by using flexible parametric geometric modeling and multi-disciplinary simulation modeling techniques (e.g., two-level variable structure, modular programming). As a result, a new MOO model consisting of a new set of performance objectives, constraints and design variables is achieved.

  The above three groups of general actions can iterate once or multiple times. At the end of the re-formulation iteration(s), a final MOO model consisting of a final set of performance objectives, constraints and design variables is achieved.

- **Phase-III:** Optimization Problem Solving (OPS)

  This phase deals with the solving of the final MOO model. It involves two groups of general actions.

MOO setup and execution: samples can be selected manually from the design space defined by the final MOO model (more precisely, from the high-performing clusters of samples in that design space). The chosen samples, as an initial population for optimization, are used to generate qualitative data sets and quantitative data sets automatically according to an optimization algorithm, thus achieving optimization results.

MOO result comparison: designers can compare the optimization results for different purposes, such as for making final design decisions and/or for verifying factors that may affect the behaviors of the used method. Thus, the final output can be Pareto solutions and/or knowledge about the used method.

The proposed method has two subtypes that are distinguished from each other mainly by their different numbers of re-formulation iterations.

- **Subtype-I:** Non-dynamic, Interactive Re-formulation method

  This subtype method (exemplified in FIG.3.4) includes one re-formulation iteration. It can suit the design context where the main purpose is to reduce existing design possibilities (i.e., shrink exploration space), such as the circumstance in the relatively late sub-phase of conceptual architectural design.

- **Subtype-II:** Dynamic, Interactive Re-formulation method

  This subtype method (exemplified in FIG.3.5.) includes multiple re-formulation iterations. It can suit the design context where the main purpose is to spark new design possibilities (i.e., expand exploration space), such as the circumstance in the relatively early sub-phase of conceptual architectural design.

  In a single re-formulation iteration, specific actions can be customized and different computational techniques can be adopted. For example, in FIG.3.5, design concepts can be re-formulated convergently and/or divergently; different data analysis techniques can be used to extract different kinds of information.

4 **How to select software tools and integrate them seamlessly into the proposed software workflow?**

This sub-question (discussed in Chapter 4) relates to the sub-goal of establishing a software workflow that can support the implementation of the proposed optimal-design method.

Visual Programming (VP) software and Process Integration and Design Optimization (PIDO) software have the potential to form a desired software workflow. In this research, McNeel's Grasshopper (with simulation plug-ins for Daysim, EnergyPlus, and Karamba3D) and ESTECO's modeFRONTIER have been selected, given that they can offer computational techniques necessary for the proposed method.

The integration of Grasshopper and modeFRONTIER can be achieved by using an integration plug-in. Given the limitations of the old plug-in, a new one (i.e., Gh-mF node) has been developed based on the collaboration between the Chair of Design Informatics at TU Delft and ESTECO SpA. The new plug-in has been through several rounds of development so that it can improve communication initiation, stabilize automatic data exchange, and simplify integration preparation. Eventually, this development has led to a formally supported direct integration node in modeFRONTIER.

5 **How to demonstrate the use of the proposed optimal-design method and verify its benefits and associated affecting factors through case studies concerning indoor sports halls?**

This sub-question (discussed in Chapter 5 and 6) relates to the sub-goal of providing case studies that can be used to establish the validity of the proposed optimal-design method.

With the aid of the proposed software workflow and the new integration plug-in, two case studies concerning indoor sports halls have been conducted to demonstrate the use of the proposed method and verify its benefits and associated affecting factors.

Case Study I has considered the conceptual design of the overall geometry of a sports competition hall, in the context that highlights reducing existing design possibilities. The Subtype-I method (i.e., non-dynamic method) has been applied; and it has focused on a one-time re-formulation process that concerns mainly removing existing variables (i.e., refining an existing concept convergently).

Case Study II has considered the conceptual design of the skylight geometry of a sports training hall, in the context that highlights sparking new design possibilities. The Subtype-II method (i.e., dynamic method) has been applied; and it has focused on a three-time re-formulation process that concerns mainly adding new variables (i.e., enriching new concepts divergently).

At the end of each case study, the benefits of adopting the subtype method and the factors that affect its behaviors have been verified by using comparative analysis. Moreover, reflections have been made to further understand the subtype method.

7.3 Limitations and future recommendations

In this section, the research limitations and future recommendations are described. They relate to issues concerning the method development (Section 7.3.1) and the method application (Section 7.3.2).

7.3.1 Issues concerning the method development

A possible bottleneck that can limit the potential of the proposed method is a large amount of time invested in parametric modeling. Indeed, parametric models are powerful for creating various geometric variations. Nevertheless, the modeling process itself is usually time-consuming. In many cases, it is not the fault of parametric tools, but of the abuse of these tools. It is easy for parametric designers to dive too deep into the fascinating details of one particular design concept, forgetting to jump out of the box. This is contradictory to the need to explore new concepts divergently during conceptual architectural design.

To solve this issue, parametric designers should focus more on the proper ways of using the tools to create parametric models that allow easy modification across different design concepts, for instance, adopting techniques like two-level variable structure, modular programming and others.

Another possible barrier that can limit the potential of the proposed method is a large amount of time invested in running high-fidelity simulation models. The time needed for running such simulation models usually accounts for the major portion of the time needed for running the whole optimization process. Reducing the simulation time can help to improve the efficiency of the overall optimization process, facilitating the use of the proposed method in practice.

For reducing the simulation time, a promising future research direction can be replacing high-fidelity simulation models with surrogate models during the optimization running process. The surrogate models are constructed by using data drawn from high-fidelity models, and they can provide fast approximations of performance objectives and constraints (Queipo et al., 2005). The potential of using surrogate models in architectural design optimization has been preliminarily shown in some of the author's previous publications (Yang et al., 2016a; Yang et al., 2016b). It has been found that surrogate models can locate the promising

area of objective space efficiently, but the accuracy of surrogate models needs to be improved locally. Moreover, computational tools that utilize advanced machine learning techniques to create surrogate models are also on rapid development, such as a Grasshopper plug-in called Opossum (Wortmann, 2017).

7.3.2 Issues concerning the method application

The two case studies in this research only include the "*internal verification*" rather than the "*external validation*" of the proposed method.

According to the Oxford dictionary, verification is "*the process of establishing the validity of something*", while validation is "*the action of checking or proving the validity of something*". In software engineering standards, verification means "*a test of a system to prove that it meets all its specified requirements at a particular stage of its development*", while validation means "*an activity that ensures that an end product stakeholder's true needs and expectations are met*" (Plutora, 2020). In this research, verification refers to a process of evaluating a work-in-progress method, to check whether the method meets the requirements specified by the internal developer at a particular stage of its development; in contrast, validation refers to a process of evaluating the end method, to check whether the method meets the external users' true needs when placed in its intended environment. Given this definition, it is necessary to carry out systematic external validation via more case studies in future research, to prove the generalization of the proposed method.

Future case studies can highlight the aspects that have not been fully studied in this research, as described below.

First, the two case studies in this research have focused more on the re-formulation of design variables, rather than the re-formulation of performance measures. They have just re-formulated performance measures once in a convergent manner. Thus, it is meaningful for future case studies to re-formulate performance measures multiple times in a divergent manner.

Second, the two case studies in this research have focused particularly on indoor sports halls, rather than other types of complex buildings. They have just studied some performance measures that are especially important for indoor sports halls. Thus, it is meaningful for future case studies to study more performance measures relevant to other types of complex buildings.

References

Plutora (2020), Verification vs Validation: Do You Know the Difference? https://www.plutora.com/blog/verification-vs-validation [Accessed: 24-September-2021].

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., & Tucker, P. K. (2005). Surrogate-based analysis and optimization. Progress in Aerospace Sciences, 41(1), 1-28. DOI: https://doi.org/10.1016/j.paerosci.2005.02.001.

Wortmann, T. (2017). Introducing and evaluating a model-based optimization tool for Grasshopper. 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 (pp. 283-293). CAADRIA and Xi'an Jiaotong-Liverpool University. ISBN: 978-988-19026-8-9.

Yang, D., Sun, Y., Di Stefano, D., Turrin, M., & Sariyildiz, S. (2016b). Impacts of problem scale and sampling strategy on surrogate model accuracy: An application of surrogate-based optimization in building design. In: 2016 IEEE congress on evolutionary computation (CEC) (pp. 4199-4207). IEEE, Piscataway. ISBN: 978-1-5090-0623-6.

Yang, D., Sun, Y., Sileryte, R., D'Aquilio, A., & Turrin, M. (2016a). Application of surrogate models for building envelope design exploration and optimization. In: R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), 2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 11-14). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

Appendices

Appendix I

Review of Multi-Objective Optimization (MOO) design methods (revised from Yang et al., 2020)

	Literature	Application field	Initial quantitative objectives and constraints	Initial qualitative objectives and constraints	Initial design variables	Ways of Re-OPF	Dy-namic Re-OPF	Inter-active Re-OPF
Type 1	Heiselberg et al. (2009)	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
Type 1	Shen and Tzempelikos (2013)	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. (2016)	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
Type 2	Curtis et al. (2013)	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
Type 2	Curtis et al. (2013)	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

>>>

Review of Multi-Objective Optimization (MOO) design methods (revised from Yang et al., 2020)

	Literature	Application field	Initial quantitative objectives and constraints	Initial qualitative objectives and constraints	Initial design variables	Ways of Re-OPF	Dy-namic Re-OPF	Inter-active Re-OPF
Type 3	Brintrup et al. (2007)	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 (2015)	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. (2011)	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 (2010)	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

>>>

Review of Multi-Objective Optimization (MOO) design methods (revised from Yang et al., 2020)

	Literature	Application field	Initial quantitative objectives and constraints	Initial qualitative objectives and constraints	Initial design variables	Ways of Re-OPF	Dy-namic Re-OPF	Inter-active Re-OPF
Type 4	Newton (2018)	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 (2013)	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

Appendix II

Review of Multi-Objective Optimization (MOO) software workflows (revised from Yang et al., 2020)

	Literature	Tool	Six kinds of techniques (implemented using the tool)					
			(1) Parametric geometric modeling	(2) Multi-disciplinary simulation modeling	(3) Multi-objective optimization algorithms	(4) Sampling algorithms	(5) Quantitative data analysis	(6) Qualitative data visualization
Type 1	Caldas (2001, 2006, 2008) Caldas and Norford (2002, 2003)	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. (2014)	--	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. (2006)	--	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 (2013) Conti et al. (2015)	--	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 (2010)	--	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 (2012, 2014)	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 Software Ltd.	DesignBuilder	Fast modeling 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)

>>>

Review of Multi-Objective Optimization (MOO) software workflows (revised from Yang et al., 2020)

	Literature	Tool	Six kinds of techniques (implemented using the tool)					
			(1) Parametric geometric modeling	(2) Multi-disciplinary simulation modeling	(3) Multi-objective optimization algorithms	(4) Sampling algorithms	(5) Quantitative data analysis	(6) Qualitative data visualization
Type 2	Janssen et al. (2011), Janssen (2013, 2015)	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)
Type 2	Von Buelow (2012, 2016)	ParaGen	Visual programming (in Generative Componens)	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)
Type 2	Vierlinger and Bollinger (2014), Negendahl and Nielsen (2015)	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)
Type 2	Brown and Mueller (2016), Brown et al. (2016)	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)
Type 3	Flager et al. (2009b)	--	BIM with limited parametric capabilities (in Digital Project)	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)

>>>

Review of Multi-Objective Optimization (MOO) software workflows (revised from Yang et al., 2020)

	Literature	Tool	Six kinds of techniques (implemented using the tool)					
			(1) Parametric geometric modeling	(2) Multi-disciplinary simulation modeling	(3) Multi-objective optimization algorithms	(4) Sampling algorithms	(5) Quantitative data analysis	(6) Qualitative data visualization
Type 4	ESTECO SpA.	GH-MF (prototype)	Visual programming (in Grasshopper)	Daylight simulation (in Daysim) Energy, thermal simulation (in EnergyPlus) 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)

Appendix III

Review of the conceptual sports building design that applies Multi-Objective Optimization (MOO), Single-Objective Optimization (SOO), or no optimization methods

Literature	Building types		Building geometries			Building performances									Optimization		
	Outdoor stadium	Indoor stadium	Grand-stands	Building envelopes	Roof structures	View quality	Solar radiation	Day-lighting	Thermal	Operational energy	Structural	Embodied energy	Acoustic	Others	No	SOO	MOO
Sun et al. (2013)	x		x			x									x		
Joseph et al. (2015)	x		x			x									x		
Zargar and Alaghmandan (2019)	x		x			x										x	
Bianconi et al. (2020)	x		x			x										x	
Shi and Yang (2013)	x			x			x									x	
Zhao and Mei (2013)		x		x				x							x		
Ding (2017)		x		x				x							x		
Heinzelmann (2018)		x		x				x								x	
Rajagopalan and Luther (2013)		x		x					x						x		
Cheng et al. (2016)		x		x					x						x		
Suo et al. (2015)		x		x					x						x		
Nord et al. (2015)		x		x					x						x		
Josa et al. (2020)		x			x			x			x		x	x	x		
Arkinstall and Carfrae (2006)		x			x						x					x	
Holzer et al. (2007)	x				x						x					x	
Flager et al. (2009b)	x				x						x					x	
Brown and Mueller (2016)		x		x	x					x		x					x
Yang et al. (2018)		x	x	x	x			x		x	x						x
Pan et al. (2019)		x	x	x	x	x					x		x				x

Appendix IV

The actions, computational techniques and software involved in the re-formulation phase (revised from Yang et al., 2018)

Types of actions		Types of computational techniques		Types of software	
Data generation (Action C)	Sample selection	Design of experiments sampling	Uniform Latin Hypercube (ULH) Sampling *	modeFRONTIER (mF)	mF's DoE node
	Automated geometry generation Automated performance simulation	Process integration (i.e., geometry generation and performance simulation process integration)	Custom System-to-System Integration *		mF's software integration plug-in (developed in this research)
Information and knowledge extraction (Action D)	Quantitative information extraction	Quantitative data analysis (e.g., correlation analysis, cluster analysis, sensitivity analysis, summary statistics, etc.)	Self-Organizing Map (SOM) * Pearson Correlation	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)
			Smoothing Spline Analysis of Variance (SS-ANOVA) *		mF's sensitivity analysis tool
			Five-Number Summary		mF's distribution analysis chart
	Qualitative information extraction Information interpretation and 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	Parametric geometric modeling	Two-Level Variable Structure *	Grasshopper (Gh)	Gh's Python script editor
			Geometry Modular Programming *		Gh's group and cluster features
	Simulation model modification Geometry-simulation model integration	Multi-disciplinary simulation modeling	Simulation Modular Programming *		
			Integrated Dynamic Modeling		Gh's simulation plug-ins (e.g., Ladybug, Honeybee, Karamba3D)

Note: "" marks the computational techniques focused on in this research.*

Appendix V

Comparison of the optimization results in Case Study I (Yang et al., 2018).

Purposes of study	MOO models	Pareto solution number	Quantitative performances of the Pareto solutions				Qualitative performances of the Pareto solutions		Unfeasible design number	Broken design number
			EUI	Mass	UDI_{mod-65}	UR_{mod}	Geometric preference	Geometric similarity		
Traditional method	MOO model 0	65	52.60 (1.30)	204.00 (38.50)	39.40 (21.20)	0.59 (0.07)	RoofSteps = 2,3,4,5	*	50 (10.8%)	21 (Con_UR_{mod}) 21 (Con_SC)
Proposed method	MOO model 1	26	52.40 (0.60)	191.50 (33.00)	56.10 (7.60)	0.62 (0.00)	RoofSteps = 3	***	83 (17.9%)	72 (Con_UR_{mod}) 10 (Con_SC)
Factor 1	MOO model 2	4	52.65 (1.05)	179.00 (10.50)	46.25 (7.55)	0.65 (0.02)	RoofSteps = 3	***	90 (19.9%)	67 (Con_UR_{mod}) 33 (Con_UDI_{mod-65}) 11 (Con_SC)
Factor 2	MOO model 3	40	52.10 (0.85)	178.50 (28.50)	47.75 (34.05)	0.60 (0.02)	RoofSteps = 2	**	144 (31.6%)	97 (Con_UR_{mod}) 72 (Con_UC)
	MOO model 4	20	53.10 (1.15)	214.00 (34.50)	38.65 (9.85)	0.61 (0.04)	RoofSteps = 4	**	186 (41.4%)	168 (Con_UR_{mod}) 32 (Con_SC) 10 (Con_DC)
Factor 3	MOO model 5	19	53.20 (0.85)	204.00 (62.25)	34.80 (35.55)	0.59 (0.01)	RoofSteps = 3	**	302 (65.2%)	294 (Con_UR_{mod}) 37 (Con_SC)
Factor 4	MOO model 1	29	53.00 (0.93)	215.00 (79.25)	43.90 (17.07)	0.62 (0.03)	RoofSteps = 3	**	165 (35.3%)	154 (Con_UR_{mod}) 26 (Con_SC)

In the 4th, 5th, 6th, 7th column, median performance values and interquartile ranges are shown without and with parentheses respectively.

In the 9th column, the number of stars represents the degree of geometric similarity; the more stars the more similar.

In the last column, major broken constraints (i.e., those violated by more than 10 designs) are listed; some designs may violate multiple constraints.

Curriculum Vitae

Ding Yang

Personal Information

Nationality
Chinese

Languages
Mandarin, Cantonese, English

E-mail
52503028@qq.com, yd52503028@gmail.com

Research Interests

Ding Yang's research interest is computational design, specifically architectural design optimization. He advocates using building performance as a guiding principle in architectural design, and has been focusing on the use of multi-objective optimization and multi-disciplinary optimization in the conceptual design of complex buildings like sports halls. Different computational techniques can be involved in his research, including parametric modeling, performance simulation, optimization algorithms, sampling algorithms, advanced data analysis, surrogate models, etc. By using these techniques, he hopes to find a proper performance-based design method for improving the overall performances of buildings. Moreover, he is also interested in adapting the method and relevant tools for the design of a wider range of building types and for urban design.

Educational Background

09.2003 – 03.2005	**Undergraduate in Finance** South China University of Technology (SCUT), China
04.2005 – 07.2009	**Undergraduate in Architecture (B.Arch.)** South China University of Technology (SCUT), China

09.2009 – 07.2011	**Master student in Architectural Design and Theory** South China University of Technology (SCUT), China
09.2011 –	**Ph.D. student in Architectural Design and Theory** South China University of Technology (SCUT), China Supervised by: Prof.dr.ir. Y. Sun
01.2014 –	**Ph.D. student in Design Informatics** Delft University of Technology (TU Delft), the Netherlands Supervised by: promotors Prof.dr.ir. I.S. Sariyildiz and Prof. dr.ir. Y. Sun and copromotor Dr. M. Turrin

Professional Experience

03.2008 - 11.2008	**Architect intern** Guangzhou Branch, CAPOL International & Associates Group, China Primary tasks: conceptual design of residential buildings, educational buildings, sports buildings
07.2009 – 01.2014	**Architect** Sun Yimin Studio, Architectural Design & Research Institute of SCUT, China Primary tasks: conceptual design and design development of sports buildings, educational buildings, cultural building and other types of large public buildings
09.2010 – 07.2011	**Teaching Assistant** School of Architecture, South China University of Technology, China Primary tasks: assisting the teaching and coordination of undergraduate courses
01.2012 – 12.2016	**Researcher** Project name: Decision-making and Design Research of Large-scale Public Buildings Based on Sustainability Granted by: National Natural Science Foundation of China (NSFC), China
2015 – 2020	**Researcher** (1) Project name: Computational Performance-oriented Design for Energy Efficiency - Case Study in Subtropical Climate Granted by: State Key Laboratory of Subtropical Building Science, China

(2) Project name: Multi-objective Optimization of Sports Building Envelope
Granted by: TU Delft Sports Engineering Development Fund, the Netherlands
(3) Project name: Multi-objective Multidisciplinary Optimization of Structural and Building Envelope Design
Granted by: Arup Amsterdam, the Netherlands

2016 – 2017 **Co-developer**
ESTECO, Trieste, Italy
Primary tasks: assisting the development of the Grasshopper-modeFRONTIER plug-in; establishing multi-objective and multi-disciplinary optimization software workflow suitable for conceptual architectral design

2020 – **Architect**
Sun Yimin Studio, Architectural Design & Research Institute of SCUT, China
Primary tasks: conceptual design, design development and contruction drawings of sports buildings

2022 – **Researcher**
Project name: Performance-based Design Theory, Method and Technology of Large-scale Public Buildings' Roof Systems
Granted by: National Natural Science Foundation of China (NSFC), China

Academic Activities

05.2015 **Presenter**
2015 IEEE Congress on Evolutionary Computation (CEC)
Sendai, Japan

08.2015 **Presenter**
IASS 2015 Amsterdam Symposium: Future Visions
Amsterdam, the Netherlands

05.2016 **Presenter**
The 7th Annual Symposium on Simulation for Architecture and Urban Design (SimAUD)
London, UK

07.2016 **Presenter**
2016 IEEE Congress on Evolutionary Computation (CEC)
Vancouver, Canada

11.2016	**Presenter and collaborator** Workshop on developing the Grasshopper-modeFRONTIER plug-in ESTECO, Trieste, Italy
12.2016	**Reviewer** The 22nd Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2017) Suzhou, China
02.2017	**Presenter and instructor** Workshop on using the Grasshopper-modeFRONTIER plug-in TU Delft, Delft, the Netherlands
05.2017	**Presenter** The 8th Annual Symposium on Simulation for Architecture and Urban Design (SimAUD) Toronto, Canada

Awards & Scholarships

2014	**Joint Ph.D. Program Scholarship** Awarded by: China Scholarship Council
2016	**Joint Ph.D. Program Scholarship** Awarded by: South China University of Technology
05.2016	**2016 Best Paper Award** Awarded by: Symposium on Simulation for Architecture and Urban Design Paper title: Supporting Exploration of Design Alternatives using Multivariate Analysis Algorithms Paper link: http://www.simaud.org/proceedings/
11.2017	**2017 Excellent Architectural and Engineering Design Award (First Prize)** Awarded by: China Engineering and Consulting Association Project name: Huai'an Sports Center, Jiangsu, China Project link: http://www.scutad.com.cn/index.php?a=show&m=Product&id=101

02.2018	**2017 Science and Technology Progress Award (Second Prize)** Awarded by: Ministry of Education, the People's Republic of China Project name: Sustainable Building Technology of Large-scale Public Buildings in Subtropical Climate Zone Project link: http://www.moe.gov.cn/srcsite/A16/s7062/201802/t20180227_327879.html
08.2021	**2021 Excellent Architectural and Engineering Design Award (Second Prize)** Awarded by: China Engineering and Consulting Association Project name: Wuhan University Sports Hall, Hubei, China Project link: http://www.scutad.com.cn/index.php?a=show&m=Product&id=104
03.2022	**2021 Science and Technology Progress Award (First Prize)** Awarded by: People's Government of Guangdong Province Project name: Sustainable Building Technology and Its Application in Large-scale Sports Buildings Project link: https://www.gd.gov.cn/zwgk/wjk/qbwj/yf/content/post_3912304.html

Publications

Journal articles

2018

Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. Automation in Construction, 92, 242-269. DOI: https://doi.org/10.1016/j.autcon.2018.03.023.

2020

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, 103251. DOI: https://doi.org/10.1016/j.autcon.2020.103251.

Conference papers

2015

Yang, D., Turrin, M., Sariyildiz, S., & Sun, Y. (2015). Sports building envelope optimization using multi-objective multidisciplinary design optimization (M-MDO) techniques: Case of indoor sports building project in China. In: 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 2269-2278). IEEE, Piscataway. ISBN: 978-1-4799-7492-4.

Yang, D., Sun, Y., Turrin, M., Buelow, P. V., & Paul, J. (2015). Multi-objective and multidisciplinary design optimization of large sports building envelopes: a case study. In: Proceedings of IASS Annual Symposia, IASS 2015 Amsterdam Symposium: Future Visions – Computational Design (pp. 1-14). IASS. ISSN: 2518-6582.

2016

Yang, D., Sun, Y., Sileryte, R., D'Aquilio, A., & Turrin, M. (2016). Application of surrogate models for building envelope design exploration and optimization. In: R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), 2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 11-14). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

D'Aquilio, A., Sileryte, R., Yang, D., & Turrin, M. (2016). Simulating natural ventilation in large sports buildings: prediction of temperature and airflow patterns in the early design stages. In: R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), 2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 43-50). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

Sileryte, R., D'Aquilio, A., di Stefano, D., Yang, D., & Turrin, M. (2016). 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 (pp. 215-222). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-05872-1.

Yang, D., Sun, Y., Di Stefano, D., Turrin, M., & Sariyildiz, S. (2016). Impacts of problem scale and sampling strategy on surrogate model accuracy: An application of surrogate-based optimization in building design. In: 2016 IEEE congress on evolutionary computation (CEC) (pp. 4199-4207). IEEE, Piscataway. ISBN: 978-1-5090-0623-6.

Turrin, M., Yang, D., D'Aquilio, A., Sileryte, R., & Sun, Y. (2016). Computational design for sport buildings. Procedia engineering, 147, 878-883. DOI: https://doi.org/10.1016/j.proeng.2016.06.285.

2017

Yang, D., Sun, Y., di Stefano, D., & Turrin, M. (2017). A computational design exploration platform supporting the formulation of design concepts. In: M. Turrin, B. Peters, W. O'Brien, R. Stouffs, & T. Dogan (Eds.), 2017 Proceedings of the Symposium on Simulation for Architecture and Urban Design (pp. 35-42). The Society for Modeling and Simulation International, San Diego. ISBN: 978-1-365-88878-6.

22#25

Design as Exploration

Multi-Objective and Multi-Disciplinary Optimization (MOMDO) of Indoor Sports Halls

Ding Yang

There are an increasing number of optimal-design paradigms used in architectural design nowadays. In these paradigms, a design task is formulated, or partially formulated, as an optimization problem. Multi-Disciplinary Optimization and Multi-Objective Optimization, as two important optimal-design paradigms, have shown their great potential in improving the performances of complex buildings in recent decades. Nevertheless, current paradigms for ill-defined conceptual architectural design still lack ways to ensure the achievement of a reliable optimization problem, which hinders reliable design solutions despite the use of advanced optimization algorithms.

To address this problem, it is necessary to shift the focus from Optimization Problem Solving to Optimization Problem Formulation. This research particularly focuses on knowledge-supported, dynamic and interactive Optimization Problem Re-Formulation in order to construct a new Multi-Objective and Multi-Disciplinary Optimization (MOMDO) method suitable for use in ill-defined conceptual architectural design. The proposed method consists of two subtype methods: Non-dynamic, Interactive Re-formulation method (Subtype-I) and Dynamic, Interactive Re-formulation method (Subtype-II), which can be used to explore design space in a convergent and divergent manner respectively. To support the re-formulation, various kinds of information and knowledge need to be extracted by utilizing different computational techniques, such as advanced sampling algorithms, Self-Organizing Map, Hierarchical Clustering, Smoothing Spline Analysis of Variance, Two-Level Variable Structure and modular programming. Moreover, a software workflow that can provide these computational techniques is developed; it integrates McNeel's Grasshopper, ESTECO's modeFRONTIER and simulation software tools Daysim, EnergyPlus and Karamba3D. With the support of this software workflow, the proposed method is demonstrated via two case studies concerning the conceptual design of indoor sports halls.

A+BE | Architecture and the Built Environment | TU Delft BK