



# UNIVERSITY OF CALGARY

**University of Calgary**

**PRISM: University of Calgary's Digital Repository**

---

Graduate Studies

The Vault: Electronic Theses and Dissertations

---

2015-05-01

## Optimizing Creatively in Multi-Objective Optimization

Ashour, Yassin

---

Ashour, Y. (2015). Optimizing Creatively in Multi-Objective Optimization (Unpublished master's thesis). University of Calgary, Calgary, AB. doi:10.11575/PRISM/27214  
<http://hdl.handle.net/11023/2211>  
master thesis

---

University of Calgary graduate students retain copyright ownership and moral rights for their thesis. You may use this material in any way that is permitted by the Copyright Act or through licensing that has been assigned to the document. For uses that are not allowable under copyright legislation or licensing, you are required to seek permission.

*Downloaded from PRISM: <https://prism.ucalgary.ca>*

UNIVERSITY OF CALGARY

Optimizing Creatively in Multi-Objective Optimization

by

Yassin Salah El-Din Ashour

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF MASTER OF ENVIRONMENTAL DESIGN

GRADUATE PROGRAM IN ENVIRONMENTAL DESIGN

CALGARY, ALBERTA

APRIL, 2015

© Yassin Salah El-Din Ashour 2015



## **Abstract**

Designers will always face the challenge of designing well-performing buildings using what are often conflicting and competing performance objectives. This thesis presents a workflow called the ‘creative optimization workflow’ using a Multi-Objective Optimization (MOO) engine called Octopus that runs within Grasshopper3D, a parametric modeling tool, and multiple simulation software. Developed as part of the workflow, the ‘creative optimization tools’ make MOO based solutions more accessible to the designer and enables the exploration and analysis of the solutions from a visual and analytical point of view. Through the juxtaposition of extreme performing solutions, serendipity is created and the potential for better multiple performing solutions is increased. The application of the workflow on the retrospective design of two buildings, De Rotterdam and the Bow Tower, is presented. The developed workflow and tools help to reduce design latency through automated generation of design solutions, incorporate disparate performance domains including formal aesthetics and explore trade-offs of multiple solutions all within one platform.

## Acknowledgements

I would first like to thank my thesis supervisor Prof. Branko Kolarevic, who offered me the opportunity to pursue this research and continued to provide insight, encouragement, personal guidance and support throughout the endeavor. He has always provided me with the materials required for the success of my studies, as well as encouraged me to look at new ideas and technology. Branko, I will be forever indebted to you past, present and future for your unwavering and generous support throughout my studies, you have been a great mentor.

I warmly thank Prof. Cormack Gates for his infinite support and guidance during the initial stages of formulating my thesis. His involvement and support in helping direct my research and build a foundation for it was incredibly valuable.

An endeavor of this nature and magnitude cannot be attempted without the support of a wide cast of family and friends. First I would like to thank my mom, Souzy Borai, who with her unconditional love, support and encouragement throughout my life has allowed me to pursue what I love. My ‘little’ sister, Sarah, who always teases me for working too much and reminding me that there is a home even when live half way across the world.

Mostafa Mobasher, my ‘brother’, a genuine friend who shared with me all the moments that led to this thesis. From back home, where it all began with a “Green Building Report” to us “achieving greatness”, this journey would not be the same without you – thank you my dear friend. My great friend, Ahmed Saeed who would forcefully remove me from in front of the computer and remind me that there is more to life than “spaghetti” aka Grasshopper. My great friend and ‘break buddy’, Konstantin Milchin, who listened to my endless mundane stories of my research work in optimization in design - ありがとう. My first friends in Calgary, Greg, Jen and Jess with whom the first few months of school were made to feel like home and whom were there till the end – thank you.

I want to express my deepest and sincere gratitude and appreciation to EVDS for the financial support I received throughout my studies. The entire EVDS experience is definitely a unique one and would not have been the same without you.

Finally, I would like to express my gratitude to everyone who supported me in bringing this thesis to light and for all your contributions to my work and life.

## Table of Contents

Abstract.....	ii
Acknowledgements .....	iii
List of Tables .....	vii
List of Figures and Illustrations.....	viii
Chapter 1: Introduction.....	1
1.1. Background .....	1
1.2. Research Statement .....	3
1.3. Goals and Objectives.....	4
1.4. Structure and Method.....	6
1.4.1. Logic of Inquiry .....	7
1.5. Conclusion.....	8
Chapter 2: Multi-Objective Optimization in Design .....	9
2.1. Overview .....	9
2.2. Performance-Based Design .....	10
2.3. Optimization in Design .....	12
2.4. Multi-Objective Optimization .....	13
2.4.1. Pareto-Based Optimization .....	16
2.5. Precedents.....	17
2.5.1. Conclusion .....	27
2.6. Summary .....	27
Chapter 3: Creative Optimization Workflow/Tool.....	28
3.1. Introduction .....	28
3.2. Conceptual Idea.....	29
3.3. Creative Optimization Workflow Documentation .....	30

3.3.1.	Parametric Model Setup.....	32
3.3.2.	Octopus .....	32
3.3.2.1.	Search Algorithms.....	35
3.3.2.2.	Filter strategies .....	36
3.3.2.3.	Preference search.....	38
3.3.3.	Simulation Tools.....	39
3.3.4.	Representation & Manipulation of results (Creative Optimization Tools).....	39
3.4.	Summary .....	45
Chapter 4: Case Studies.....		47
4.1.	Overview .....	47
4.2.	Case Study 1: De Rotterdam .....	47
4.2.1.	Model overview: the design loop.....	49
4.2.2.	The parametric model .....	50
4.2.3.	Analysis of results.....	52
4.3.	Case Study 2: The Bow Tower .....	57
4.3.1.	Model overview: the design loop.....	60
4.3.2.	The parametric model .....	62
4.3.3.	Analysis of results.....	64
4.4.	Analysis and Discussion.....	76
4.5.	Analysis Conclusion.....	77
Chapter 5: Conclusion and future work.....		80
5.1.	Conclusion.....	80
5.1.1.	Scope and limitations.....	82
5.2.	Future Work .....	83
5.3.	Summary .....	85

Bibliography .....	87
Appendix A.....	94
Appendix B.....	97
Appendix C.....	101
Appendix D.....	109

## List of Tables

Table 4-1. MOO properites of the De Rotterdam Tower optimization runs. ....	56
Table 4-2. Comparison of De Rotterdam optimization runs vs De Rotterdam replica.....	57
Table 4-3. Objectives under evaluation for each optimization run.....	60
Table 4-4. MOO properties of the original Bow Tower optimization runs. ....	65
Table 4-5. MOO properties of the modified Bow Tower optimization runs. ....	71
Table 4-6. Comparison of Bow Tower optimization runs vs Bow Tower replica (MG: Modified Geometry). ....	75
Table 10-1 De Rotterdam 1 <sup>st</sup> optimization run (bevel iteration) results. ....	109
Table 10-2. De Rotterdam 2 <sup>nd</sup> optimization run (stacked block iteration) results. ....	109
Table 10-3. Bow Tower 1 <sup>st</sup> optimization run (6 objectives) results. ....	109
Table 10-4. Bow Tower 2 <sup>nd</sup> optimization run (4 objectives) results.....	109
Table 10-5. Bow Tower 3 <sup>rd</sup> optimization run (3 objectives) results. ....	110
Table 10-6. Bow Tower 4 <sup>th</sup> optimization run (4 objectives) results. ....	110
Table 10-7. Bow Tower 5 <sup>th</sup> optimization run (3 objectives) results. ....	110
Table 10-8. Bow Tower 6 <sup>th</sup> optimization run (2 objectives) results. ....	110
Table 10-9. Bow Tower 7 <sup>th</sup> optimization run (1 objective) results.....	111

## List of Figures and Illustrations

Figure 1-1. As the design problem becomes more complex (from left to right), the surface describing the design space becomes more complex. Finding the optimum the optimum solution becomes more challenging (Benjamin, 2012). .....	6
Figure 2-1. Interactivator: Networked Evolutionary Design System by Frazer (1995). .....	12
Figure 2-2. Flow chart for the GA process. ....	15
Figure 2-3. Pareto optimum (non-dominated solutions) formed by the red dots and dominated solutions formed by the blue dots. ....	17
Figure 2-4. Pareto front for a case study generated by GENE_ARCH, with best solution for heating (1) visualized on the left and best solution for lighting (7) on the right (Caldas, 2007)..	19
Figure 2-5. Iterations of the algorithm optimizing a sample apartment complex building with the objectives of minimizing annual cost and wasted space (Michalek & Papalambros, 2002). ....	20
Figure 2-6. Morph-process GUI, one of the four components of system (Malkawi, 2003). ....	21
Figure 2-7. A set of optimized cantilever trusses with different unique spans generated by eifForm for a stadium roof model created in Generative Components (GC) (Shea et al., 2003). 22	
Figure 2-8. Several design iterations of the Living Light project (Koeugh & Benjamin, 2010)..	23
Figure 2-9. The five basic phases of the ParaGen cycle: select, breed, generate, evaluate, store data (Von Buelow, 2011). ....	23
Figure 2-10. Workflow applying a priori use of weights to bias the solutions before the optimization process (Cunha et al., 2011). ....	25
Figure 2-11. An example of one of the case studies explored by Cunha et al. (2011). ....	25
Figure 2-12. H.D.S. Beagle system architecture: Autodesk Revit is used for parametric generative purposes in combination with GBS for energy analysis and Microsoft Excel for the financial model (Gerber et al., 2012). ....	26
Figure 2-13. An example of a sub-set of the solution space for the twisting tower scenario in which the multi-objective ranks the solutions and visualizes them for ease of manual decision making (Lin & Gerber, 2013). ....	26
Figure 3-1. Filtering of solutions using modeFrontier and manual retrieval of the respective 3D solutions (Benjamin, 2012). ....	30
Figure 3-2. Creative optimization workflow. ....	31
Figure 3-3. Octopus Interface. ....	33

Figure 3-4. The red dots slowly form the Pareto front formation over the generations (from left to right).....	34
Figure 3-5. Octopus interface: (3-5.1): Hypervolume graph; (3-5.2): Parameter graph; (3-5.3): Convergence graph. ....	37
Figure 3-6. Preferred solutions selected (solutions highlighted in blue). ....	38
Figure 3-7. Color gradient used to visualize the level of the performance of the sorted solutions (red: worst performer; blue: best performer). ....	41
Figure 3-8. Subset searches are represented as red polylines. ....	42
Figure 3-9. Individual solution search using parallel plot graph and radar chart. ....	42
Figure 3-10. Subset search using the parallel plot graph and the Pareto graph. ....	43
Figure 3-11. Comparison of two solutions using the radar-based chart tool. ....	44
Figure 3-12. Selecting the solutions from a subset search based on formal qualities alone. ....	44
Figure 3-13. Exploration of the solutions forming the Pareto front. ....	45
Figure 4-1. De Rotterdam Tower.....	48
Figure 4-2. De Rotterdam parametric replica. ....	48
Figure 4-3. Left: De Rotterdam bevel iteration; Right: De Rotterdam stacked block iteration....	49
Figure 4-4. Generative model components of De Rotterdam Tower optimization runs. ....	50
Figure 4-5. Upper quartile of daylight factor of 1 <sup>st</sup> optimization run (beveled iteration). (1 <sup>st</sup> axis: FAR; 2 <sup>nd</sup> axis: Profit; 3 <sup>rd</sup> axis: Daylight factor; 4 <sup>th</sup> axis: Views).....	53
Figure 4-6. Direct relation between the highest FAR and profit that led to low daylight factor (red: worst performer; blue: best performer of daylight factor). ....	53
Figure 4-7. Pareto front results of the 2 <sup>nd</sup> optimization run (stacked block iteration). ....	55
Figure 4-8. 90 <sup>th</sup> percentile of daylight factor of the 2 <sup>nd</sup> optimization run (stacked block iteration). ....	55
Figure 4-9. The Bow Tower.....	58
Figure 4-10. The Bow Tower parametric replica.....	59
Figure 4-11. Left: Same basic geometry parameters as the original Bow Tower design; Right: Side arcs are tangent to the main arcs (modified geometry).....	59



Figure 4-12. Generative model components of the Bow Tower optimization runs.....	61
Figure 4-13. Views analysis of the Bow Tower parametric replica (red: best performance areas; blue: worst performance areas).....	63
Figure 4-14. Glare bouncing off the Bow Tower based on the sun vectors during the summer equinox.....	64
Figure 4-15. Sensitivity analysis of the 1 <sup>st</sup> optimization run (6 objectives) (1 <sup>st</sup> axis: FAR; 2 <sup>nd</sup> axis: Profit; 3 <sup>rd</sup> axis: Daylight factor; 4 <sup>th</sup> axis: Views; 5 <sup>th</sup> axis: Glare; 6 <sup>th</sup> axis: Shaded area).....	66
Figure 4-16. Pareto front results (represented by the Delaunay mesh) of 1 <sup>st</sup> optimization run (6 objectives).....	67
Figure 4-17. 99 <sup>th</sup> percentile of daylight factor of the 1 <sup>st</sup> optimization run (6 objectives) - three out of the nine solutions have an ellipse shaped floor plan. ....	67
Figure 4-18. 90 <sup>th</sup> percentile of profit of 1 <sup>st</sup> optimization run (6 objectives). ....	68
Figure 4-19. Parallel plot of the 2 <sup>nd</sup> optimization run (4 objectives) (1 <sup>st</sup> axis: Profit; 2 <sup>nd</sup> axis: Daylight factor; 3 <sup>rd</sup> axis: Views; 4 <sup>th</sup> axis: Shaded area).....	69
Figure 4-20. 90 <sup>th</sup> percentile of profit of the 2 <sup>nd</sup> optimization run (4 objectives) – the solutions show a larger range of geometric variety.....	69
Figure 4-21. 90 <sup>th</sup> Daylight Factor of the 3 <sup>rd</sup> optimization run (3 objectives) generated irregular shaped floor plans. ....	70
Figure 4-22. 90 <sup>th</sup> percentile profit of 4 <sup>th</sup> optimization run (4 objectives). ....	72
Figure 4-23. 90 <sup>th</sup> percentile profit of 5 <sup>th</sup> optimization run (3 objectives). ....	72
Figure 4-24. Original vs solution ID 254 (overall performs better than the Bow parametric replica). ....	73
Figure 4-25. 90 <sup>th</sup> percentile of daylight factor of 6 <sup>th</sup> optimization run (2 objectives): the generated designs are almost the same as the original Bow Tower. ....	73
Figure 4-26. Strange parametrically generated solutions that cannot be used for further design development even if they perform well. ....	76
Figure 7-1.Creative Optimization Tools.....	94
Figure 7-2. Form-based choice tool.....	94
Figure 7-3. Subset search and gallery display settings. ....	95
Figure 7-4. Parallel-plot coordinate graph tool.....	95

Figure 7-5. Pareto graph tool. ....	95
Figure 7-6. Radar-based chart tool.....	96
Figure 8-1. De Rotterdam parametric replica GH definition. ....	97
Figure 8-2. De Rotterdam (bevel iteration) GH definition. ....	97
Figure 8-3. De Rotterdam (stacked block iteration) GH definition. ....	98
Figure 8-4. The Bow Tower parametric replica.....	98
Figure 8-5. The Bow Tower original geometry (1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> optimization runs).....	99
Figure 8-6. The Bow Tower modified geometry (4 <sup>th</sup> , 5 <sup>th</sup> , 6 <sup>th</sup> , 7 <sup>th</sup> optimization runs). ....	99
Figure 8-7. Site constraints. ....	99
Figure 8-8. Program definition areas. ....	100
Figure 8-9. Financial performance: profit calculation (the Bow Tower).....	100
Figure 9-1. De Rotterdam (bevel iteration) Octopus results. ....	101
Figure 9-2. De Rotterdam (stacked block iteration) Octopus results.....	102
Figure 9-3. Bow Tower 1 <sup>st</sup> optimization run (6 objectives) Octopus results.....	103
Figure 9-4. Bow Tower 2 <sup>nd</sup> optimization run (4 objectives) Octopus results. ....	104
Figure 9-5. Bow Tower 3 <sup>rd</sup> optimization run (3 objectives) Octopus results. ....	105
Figure 9-6. Bow Tower 4 <sup>th</sup> optimization run (4 objectives) Octopus results. ....	106
Figure 9-7. Bow Tower 5 <sup>th</sup> optimization run (3 objectives) Octopus results. ....	107
Figure 9-8. Bow Tower 6 <sup>th</sup> optimization run (2 objectives) Octopus results. ....	108

## Chapter 1: Introduction

### 1.1. Background

The introduction of numerical methodologies such as structural analysis and the advent of computers encouraged the use of the scientific approach to problem solving in architectural problems. Through mathematicians whom later became architects and vice versa that substantial work in the field of CAAD (Computer Aided Architectural Design) grew significantly. The first algorithms were used for generating building layouts in 1960s, the shell design of the Sydney Opera House required Arup's use of analysis software (Gero & Radford, 1988) and, the first integrated package for building performance appraisal was used in 1972 (Maver, 2002). It was in the seventies that "a battery of computer aids for providing the designer with evaluative feedback on his design proposals" were developed.

A central challenge for designers is to design buildings which perform on multiple fronts (i.e. "which work – economically, socially and technically") using what are often conflicting and competing objectives (Maver, 1987). The aspiration is to make designs as good as possible – *multiple performative* and the desire to ensure that our design endeavors do indeed match our needs in the best possible way. It is difficult for a designer to fully articulate multiple aspects that influence a design especially during the conceptual design stage, where design decisions have a significant impact on the final performance of the design.

In addition, the requirements of a design come from the client, the community and the social ethos of the time - some of these requirements are physical, some psychological and some social. It is this nature of design – its non-quantifiable attributes and multi-dimensional objectives – that a designer is required to aid in optimizing these objectives. Yet, even with these obvious difficulties, architectural researchers have found optimization as a solution to many sub-problems in design (Gero, 1975). With this in mind, it needs to be stated that neither the computer nor optimization is capable of solving every aspect of the real and difficult task that is design.

Optimization is one method that designers could use to aid in the search for a 'good' solution. Optimization in design developed mainly into two directions over the past few decades: the automation of solving specific architectural problems such as building layouts and; the

development of interactive formal design tools that can be coupled with simulation tools. Specifically Multi-Objective Optimization or Multi-Criteria Design Optimization (MOO/MCDO) or Pareto optimization was developed primarily for and by industrial engineers in the aerospace and automobile industry. Pareto optimization is “more realistic and useful for design”, according to Gero & Radford (1988), “because it allows subjective criteria to be taken into account.” Compared to other optimization techniques, Pareto based optimization is currently the leading multi-objective search technique based on the work of economist Vilfredo Pareto (Davis, 2009; Benjamin, 2012). Advancements in building performance simulation software and easier coupling between generative and simulation software streamlined the path towards performance-based generative design (Kolarevic, 2004).

Current Computer-Aided Design and Engineering (CAD/CAE) tools offer more rapid design iterations via quick visualization, modification of geometry and ability to simulate many different aspects of building performance (Frazer, 1995; Burry & Murray, 1997; Aish & Woodbury, 2005; Lin & Gerber, 2013). Aesthetics, part of a building’s performance, among other subjective aspects, are naturally ignored in optimization models as they are particularly difficult to quantify mathematically (Michalek & Papalambros, 2002; Bittermann, 2009; Cunha *et al.*, 2011; Turin *et al.*, 2011; Benjamin, 2012). Manually simulating many design alternatives and analyzing them can be a very time consuming endeavor, so designers are often forced to select from a narrow set of solutions. Architectural design, however, involves a great deal of complexity and requires judgment and decision making (Coello Coello *et al.*, 2007; Bittermann, 2009).

Even with the increasing use of MCDO in the building industry, it has yet to be fully embraced as a vital part of the design process (Evins, 2013). There are several reasons for the slow assimilation of MCDO tools in design. Designers must visually identify solutions resulting from MCDO in order to fully evaluate design trade-offs along with quantifiable goals, preferences and constraints (Michalek & Papalambros, 2002; Roy *et al.*, 2008; Haymaker & Flager, 2009; Cunha *et al.*, 2011; Marsault, 2013).

Rigorous analysis and comparison of the design solutions using both quantitative and qualitative criteria becomes essential to identify ‘good/sub-optimal’ solutions thus improving decision making (Cunha *et al.*, 2011; Turin *et al.*, 2011; Benjamin, 2012). Lack of real-time analysis and

feedback latency between performance metrics and geometry in MCDO design is another obstacle that exists in the current tools which invariably leaves a significant area of the design solution space unexplored (Haymaker & Flager, 2009; Sanguinetti *et al.*, 2010). In a conventional MCDO process, the majority of the designer's time is spent managing design information in their task specific format and coordinating solutions. A pattern identified by Bradner *et al.* (2014) through interviewing practitioners that use MCDO in their design process, found that through badly designed user interfaces and inflexible data exploration and visualization tools, designers have a difficult time understanding the design solution space.

These challenges that face the industry are representative of how as tools are evolving, so is the need for the designer to become more software literate to customize his/her own tools for a more seamless integration. This evolution is part of the performative architectural style that has continually evolved over the last two decades. Performative architecture is representative of both the process of design and its adaptation of the environmental context, i.e. it is design that is flexible to its needs *a priori* and *posteriori*. MCDO enables the exploration of multiple performative aspects of a design in order to generate a 'good' design in a quick and versatile manner. It is with this main pursuit in design that inspires designers to look forward and support today's technology. With the ever growing aspiration to improve the built environment and with the ever growing conscious of designing 'sustainable' architecture, that innovation is necessary and will lead to good practice and design.

## **1.2. Research Statement**

Defining the performative objectives of a design and balancing between what are often conflicting objectives in an effective way is a central challenge for designers. For example, a design can perform very well in one particular domain (such as, for example, daylight factor performance) yet does not perform well in other domains (such as financial performance). Notably, design performativity has been at the forefront of architectural discourse mainly because of the sustainability agenda that has become part of the social and political climate. Performative design is defined as a design performing on multiple domains: quantitatively (thermal, lighting, structural, etc.) and qualitatively (economical, spatial, aesthetics, etc.) (Kolarevic, 2005). A well –performing solution representing all objectives optimized is rare,

hence, the exploration of sub-optimal designs is necessary and the study of their trade-offs is necessary.

With this in mind, one of the approaches to curtail the impacts of the destructive behavior of the current built environment is the encouragement of high-density buildings (skyscrapers). It is crucial to the design of a successful skyscraper to perform on multiple fronts for it to become more viable and arguably a more attractive investment for developers and occupants. In parallel, the design profession is currently undergoing technological advancement which will allow for faster simulation engines, improved software integration and better MCDO algorithms. This will subsequently enable rapid design exploration taking into account the formal qualities of a design, mitigation of exchange and interfacing issues and overall reduced design latency.

**The integration of MCDO with interactive information visualization tools will help improve the understanding of the quality of the design solution space, while enabling designers to evaluate the formal qualities of a design and should result in solutions that are multiple performative.** – *Hypothesis*

The hypothesis combines two aspects that aid generative design: “information visualization tools” and “MCDO”. Within the context of this thesis, 1) information visualization tools is the means of enhancing the designer’s understanding of 2) the MCDO resultant design solution space.

The statement focuses on the application of information visualization tools and its impact on MCDO based solutions being more accessible to the designer. The tools will also provide the ability for the designer to evaluate formal qualities of the design as an implicit objective in the optimization process. Qualitative attributes of the design solution space can be ascertained via both the information visualization tools and Pareto-based ranking through sensitivity analysis. Through these developed tools, the workflow is enhanced significantly as well as the ability to make more informed decisions.

### **1.3. Goals and Objectives**

One of the commonly held views of design, which is used as a basis for Computer-Aided Design (CAD), is that it is a goal seeking activity with the purpose of improving design. It is the existence of goals which makes design purposeful and necessitates decisions about the best ways

to achieve those goals (Gero, 1985). Design is also a creative process that is multi-variate in nature, has a large magnitude of solutions to any given design problem and is based on the application of heuristics learnt through personal experiences and knowledge (Gero & Radford, 1988). As soon as software tools place constraints on the design process, the designers views this as a limitation on the quality of the outcome of their creative process. Ironically, the purpose of software tools is fundamentally to aid the designer in his/her design or design process, yet very few tools are sensitive and malleable to the creative process itself.

The research goal of this thesis is to develop a workflow ('creative optimization workflow') that will make MCDO generated design solutions more accessible and to enable the evaluation of formal qualities as an implicit design objective in the optimization process. The information visualization tools ('creative optimization tools') will aid the designer in exploring and evaluating more quickly and efficiently the design solution space. The aim is to explore the trade-offs of sub-optimal solutions i.e. 'explore' and not aim to find a singular optimum solution (i.e. 'exploit') (Figure 1-1). No bias in the objectives is included as part of the optimization search from the outset to produce the widest possible range of solutions for a given design problem. Optimization trajectory can then be altered accordingly to explore alternative geometric and topological designs and bias can be integrated later on to 'exploit' rather than 'explore', hence 'optimize creatively'. The use of optimization tools in design is not to generate the highest performing solution, but rather to gain a better understanding of the design space (Bradner *et al.*, 2014). Bradner *et al.* (2014) identified that many architects and engineers use optimization at the early stages of design to *question* and *explore* design problems rather than to *solve* design problems. Not only that but design optimization tools aid designers in generating, exploring and discovering new and sometimes unexpected design solutions.

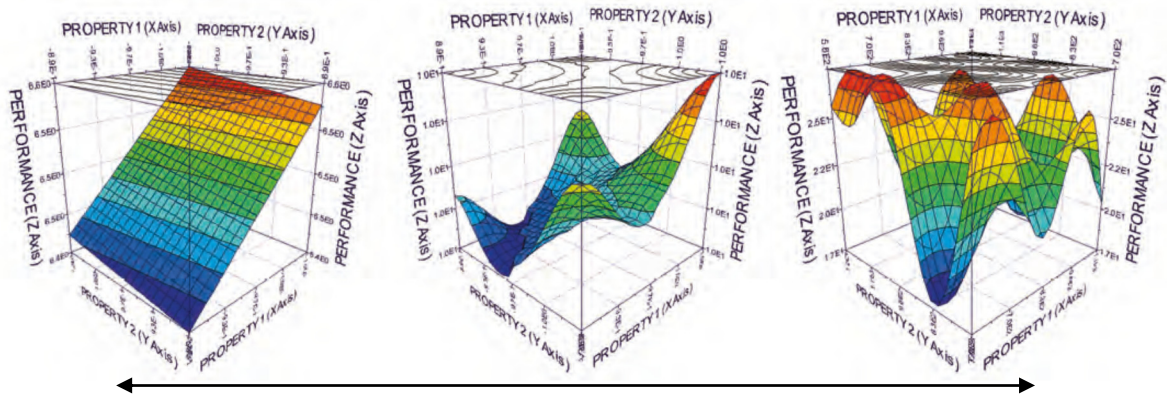


Figure 1-1. As the design problem becomes more complex (from left to right), the surface describing the design space becomes more complex. Finding the optimum the optimum solution becomes more challenging (Benjamin, 2012).

The workflow and tools are developed and tested using case studies based on already built designs to examine their impact on improving the performance of the resultant designs. The workflow does not just include simulation and analysis but allows for designer interaction and promotes a more creative exploration of the design solution space, therefore making MCDO solutions more accessible to the designer. The workflow aims to couple performance-based generative design with designer reasoning and fully engage the designer in the early stages of the design process.

This research aims to be universally applicable and the tools to be formally available to both students and practitioners. The only existent limitations of its applicability is in Grasshopper3D limitations and/or its associated plugins.

#### 1.4. Structure and Method

This thesis develops multiple interactive information visualization tools for the manipulation of the resultant data from the MCDO process. The tools are intended to be used by designers involved in the early stage design process in aiding the designer in quickly and effectively understand the design solution space and reach a desired, best or most-acceptable, yet ‘sub-optimal’ solution. The ‘sub-optimal’ solution would effectively be a solution that is multiply performative and satisfactory to the various stakeholders involved.

The study focuses on the design and influence of the developed workflow and tools on the MCDO process and subsequently on the performance of the solutions. In this study, a parametric design of two case studies is set up considering all constraints that are native to each individual case study. A MCDO process is then implemented using a Grasshopper plugin called Octopus to



generate multiple iterations based on certain performative objectives such as daylight or views performance. An analysis is carried out of the resultant design solution space using the developed information visualization tools.

#### *1.4.1. Logic of Inquiry*

Chapter 2 provides a review of MCDO and its uses in design are explored. The literature review describes existing use of MOO in architectural design and the principal reason for the use of optimization in design: the emergence of performance-based design. The second part of the chapter will review some of the recent applications of the use of optimization (some of which use MCDO) software in design.

The design of the ‘creative optimization workflow’ and the associated tools that aid the designer in exploring the design solution space are documented in Chapter 3. Overall, this workflow is dependent on the use of three main software tools: Rhinoceros3D as a modeling and visualization tool, Grasshopper3D as a parametric modeling software and Octopus for the MOO engine. In addition to these tools, several simulation tools and software are used to aid the process, such as DIVA for daylight performance or Ladybug for glare analysis, etc. Octopus, the MOO engine to be used in this study, will be examined in terms of the population and mutation strategy. The information visualization tools were developed in Grasshopper3D to allow the designer to search, sort and filter the resultant data of the evolutionary runs and make it feasible to evaluate the formal qualities of a design as an implicit objective in the optimization process.

Within this workflow, data processing depends not only on the simulation tools and their calculation capabilities, but also on the pre-defined parameters, based on the desired or required performance objectives. The evolutionary runs are not meant to stop further exploration; on the contrary, they are meant to encourage the designer to explore different formal trajectories.

In Chapter 4, the proposed workflow and associated tools are tested through two retrospective designs of the De Rotterdam Tower and the Bow Tower. The two case studies are parametrically modelled in Grasshopper3D. The developed workflow is tested on multiple iterations of the two case studies to study the influence of MCDO on design.

Chapter 5 concludes the study by summarizing the potential of the developed workflow. The scope and limitations faced by this study are discussed and solutions are suggested to address those limitations as part of future research work.

## **1.5. Conclusion**

The feasibility of applying MOO in design has become easier in the past decade with advances in software and computational power. Despite its potential, critics of MCDO argue that MOO further abstracts design into purely quantitative metrics with the aim of designing an optimized form. This thesis explores how two polarized aspects of design, quantitative and qualitative, can be coupled to work in a comprehensive manner to help develop a balanced design with ‘good’ overall performativity.

This study is dependent on the use of the already available MOO engine (Octopus) as a black box to carry out the optimization process. The methodology implemented in this study enables designers to explore multiple performance objectives of a design during the early design stage using both quantitative metrics and form performance. Moreover, the workflow integrates MOO, simulation tools and the developed information visualization tools into a design/modeling tool, which mitigates the interfacing and exchanging of data. The proposed workflow is intended to provide flexibility by allowing the possibility of exploring disparate performance domains in a meaningful manner with reduced design latency due to the automation of the generation of solutions for different design problems.

In the following chapters, an elaboration of the investigated topic is presented along with the application of the developed workflow and tools on two case studies.

## Chapter 2: Multi-Objective Optimization in Design

### 2.1. Overview

A designer's pursuit for achieving a 'well' performing design has been the motivation for the development and use of synthesis and evaluation tools. Optimization is used as a design aid in the design process to improve the performance of a design whether for a small sub-problem or an overall design generation. Evolutionary Algorithms (EA) have traditionally been used to solve optimization problems, and can also be used as design aids (Malkawi, 2005). Genetic Algorithms (GA), an EA type, first introduced by John Holland in the 1970s is a method inspired by a biological mechanism and is used for solution generation in MCDO.

Multi-Criteria Design Optimization (MCDO) has been available for more than three decades now and its application in real world problems is continuously increasing. MOO software was mainly developed for and by industrial engineers to design airplanes, automobiles and high speed trains. In the late 1980s, John Frazer, author of "An Evolutionary Architecture" began to experiment with MCDO and study its impact on morphological transformations. Other researchers such as Anthony Radford and John Gero, also in the late 1980s, developed software that used Evolutionary Algorithms (EA) for multi-criteria architectural design problems.

MCDO has since evolved significantly over the past two decades mainly due to the rapidly developing technology of computers and the improved mathematics of optimization algorithms. It has become one way of aiding designers in generating a much wider range of design possibilities that actually perform on multiple domains. The conventional design process involves postulating, evaluating and modifying potential solutions in an iterative fashion to arrive at an acceptable form. This acceptable form may not necessarily be the 'best' solution to the architectural problem. Optimization helps the designer to generate direct prescriptive information on the nature of the solutions. It also helps define what represents the best set of solutions that satisfy specified objectives by employing numerical algorithms of decision-making processes. MCDO is particularly useful for designers as it provides an opportunity to explore the implications of subjective decisions on explicit objectives (Gero & Radford, 1988). Better performing buildings that are more structurally sound with less building material or show improved quality of care in hospitals by making them more comfortable places to work, for example, are a result of the use of optimization in design (Bradner *et al.*, 2014).

This chapter will review the use of MOO in architectural design and the reason optimization in design is used - performance-based design. This review is necessary to contextualize and to explain the basis of advancement of MOO in design. The second part of the chapter will review some of the recent applications of the use of optimization (including MCDO) software in design. Lately, optimization and MCDO are pursued broadly in both academia and the profession, especially with the introduction of parametrics into design. Each case study will be briefly described in terms of the methodology and tools developed or used.

## **2.2. Performance-Based Design**

Christopher Alexander in his book *Notes on the Synthesis of Form* (1964) writes: “Every design problem begins with an effort to achieve fitness between two entities: the form in question and its context. The form is the solution to the problem; the context defines the problem.”(p.21)

Alexander was not explicitly arguing for performance-based design but was linking form with context which could also be used to describe the relation of form and performance as well.

Performance-based design is essentially an approach where the performance of a design, in terms of quantitative or qualitative objectives, is used as the guiding principle of the design process. More specifically, according to Oxman (2008): “it is the exploitation of building performance simulation for the modification of geometrical form towards the objective of optimizing a candidate design.” It is also an approach where the performance not just of form but also of materials and structure are recognized as a vital part of the design process.

Historically, performance has been an important aspect of architecture and according to Braham (2004), it draws on the notions of determinism and functionalism in architecture. These notions are evident in both Roman and Greek architecture, where both innovation and beauty are applied to provide functional and comfortable architecture (Hensel, 2008). Towards the end of the 19<sup>th</sup> Century, with the peak of the second industrial revolution, the concept of architecture fulfilling other than purely artistic purposes was established (Gero, 1975). Today architecture is conceived of as both an art and a science, the combination of form and performance (Fasoulaki, 2008).

The advent of the digital revolution in the 1950s and improved science encouraged the use of the scientific approach to problem solving in architecture (Gero, 1975). In the 1960s, Arup used computers in the analysis of the shell structure of the Sydney Opera House and, the first algorithms were used for generating building layouts (Maver, 2002; Radford & Gero, 1988). The

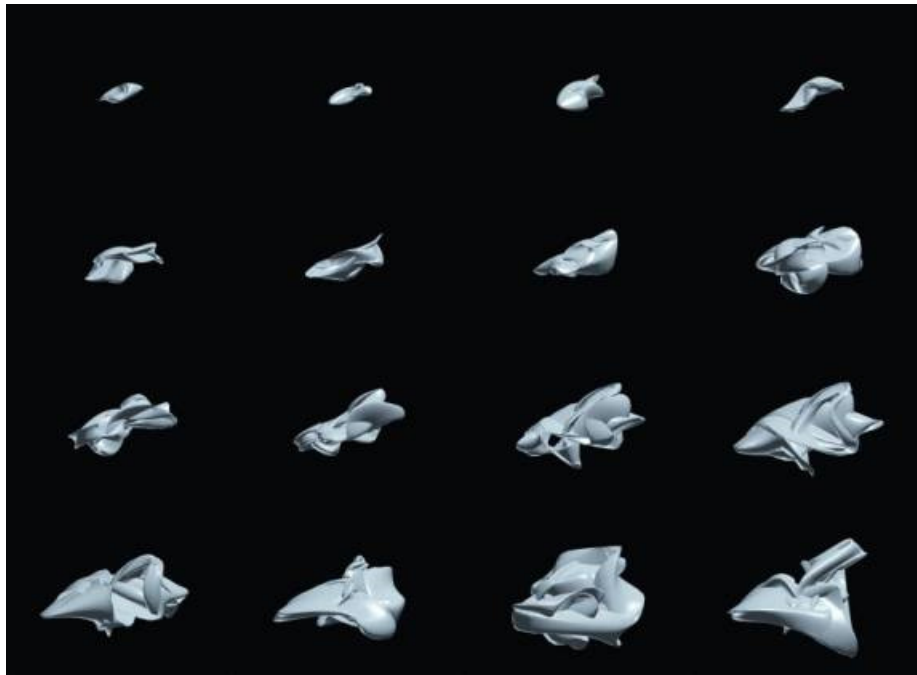
1970s resulted in a “generation of a battery of computer aids for providing the designer with evaluative feedback on his design proposals” (Cross & Maver, 1973). Significant contributions were made in terms of computer aided tools, with the first integrated package for building performance appraisal developed in 1972.

Research about building performance came to the forefront of CAAD discourse in 1972, when the Building Performance Research Unit (BPRU) at the University of Strathclyde published its first book titled ‘Building Performance’ (University of Strathclyde). The Architecture and Building Aids Computer Unit, Strathclyde (ABACUS) was founded under Thomas Maver and developed several software tools such as PACE (Package for Architectural Computer Evaluation) in 1970 as a “computer aided appraisal facility for use at strategic stages in architectural design”. Multiple digital performance analysis tools were also developed by Thomas Maver for *a priori* analysis of designs and for the purpose of aiding the designer in reaching a better design solution through man-machine interaction (Kolarevic, 2005). The tools were specific to the nature of the architectural problem, for example, PHASE (Package for Hospital Appraisal, Simulation and Evaluation) and SECS (Spatial Environment in Comprehensive Schools).

Design problems, however, cannot be polarized as qualitative (architectural) problems where they would satisfy the designers’ form aspirations, nor as quantitative (engineering) problems where the design problem is delineated into mathematical equations and is solved by algorithms (Kolarevic, 2005). Oxman (2006), describes performative design as a holistic integration of two essential processes: generation and evaluation. Along with these key essential processes is the ability to modify and manipulate digital form in response to analytical evaluation. It is the transformation of the dualism of form and performance into a synergy that leads to an integrated design solution. The increased seamless integration of multiple digital tools is also encouraging more collaboration between designers and engineers in the early stages of design (Kolarevic, 2005). This integration allowed for an evolution in the design process, from essentially the traditional process of “form making” to a process of “form finding”.

Generative design is one approach in form finding that can be applied in architectural design. In the 1980s, John Frazer was one of the first architects to apply the concept of generative design using EA and is a key pioneer in the digitization of morphological transformations (Beesley,

2006). The results of this process are visual representations which are evaluated on an encoded selection criteria (Figure 2-1). Janssen (2006) expanded and refined Frazer's work in developing a design method that generated complex three dimensional designs that are both "intelligible and unpredictable". Similar work was also explored by Shea *et al.* (2003), combining shape grammars with simulated annealing to generate discrete structures. The coupling of the design requirements (objectives) with algorithms in the early stages of the design process to produce an array of possible solutions improves the quality of a design.



*Figure 2-1. Interactivator: Networked Evolutionary Design System by Frazer (1995).*

The paradigm of performance-based design has gained momentum with the growing impact of the 'sustainable agenda'. It is with this underlying notion that generative design has provided the motivation for designers to actively engage the broader spectrum of performance-based design. It is necessary to the future development of performance-based design to apply EAs in the initial stages of the design process to enable 'exploration' of form beyond the obvious.

### **2.3. Optimization in Design**

Optimization, effectively is the search for the 'best' possible solution relative to the design problem and designer's predefined objectives. The process of optimization represents the pursuit of making something as efficient and/or effective as possible within a certain context.

Optimization uses algorithms in order to generate prescriptive information on the nature of an optimal solution through the evaluation of an explicit objective within various constraints (Gero & Radford, 1988).

Optimization tends to be used for ‘exploitation’ where it is the final ‘optimum’ solution that counts and not the generative process that occurs to achieve that result. A study carried out by Andersson (2001) outlines that different mathematical models of optimization were developed for engineering problems where the global optima is the desired result. It is with this aim that engineers are not after the novel generation of a design but rather a reliable outcome that is focused on the best quality and the least cost possible. Optimization is hereby a technical task rather than a design task; it is limited to the different autonomous specialist areas and does not enable a comprehensive approach (Vierlinger & Hofmann, 2013).

Designers recognized the potential of the morphogenetic transformations which happen during the optimization process. Pioneering research in the field of building design optimization was led by John Gero with the publishing of “Design Optimization” in 1985 and was followed by another seminal book by Anthony Radford and John Gero in 1988 titled “Design by Optimization in Architecture, Building and Construction”. The research resulted in an excellent overview and analysis of the various methods and techniques implemented in the early stages in the use of optimization in design at the time.

Design optimization adopted these optimization techniques and adapted them to architectural problems. As opposed to engineering design problems, where the objectives are clear from the outset and the final outcome is somewhat known, architectural design problems may have vague objectives from the outset with no guarantee of a ‘good’ solution. It is the need for heuristic methods that rely on a global holistic view of the problem which is more suitable for design problems (Kalay, 2004).

## **2.4. Multi-Objective Optimization**

It is the nature of design with its myriad of complex (often conflicting) objectives that by condensing these objectives into a ‘single objective’ weighted optimization equation that a design problem is not well represented. Defining the design problem into separate objectives will provide a more realistic representation of the design task and the individual objectives. Often in a

single-objective optimization problem, the search is well defined and reaching an optimized solution is easy. Yet, even with the most robust optimization tools, it is difficult to reach a solution that is optimized for multiple design objectives – both quantitative and qualitative objectives. Thus, using Evolutionary Algorithms (EA) for MCDO produces a whole set of possible solutions of equivalent quality.

The potential of using EAs was hinted at by Rosenberg in the late 1960s for solving MOO problems (Rosenberg, 1967). EAs are heuristic search methods based on the principals of natural selection which have been used in optimization problems in several fields (Deb, 2001). In the early 1970s, Rechenberg and Holland published their pioneering research on the use of EAs for optimization in the field of design where they applied the concept of Genetic Algorithms (GA) for the optimization of complex engineering problems.

EAs are very robust methods and can handle all types of fitness parameters and variables (Andersson, 2001). Each parameter is coded into a gene as a string of bits. All the parameters together form a chromosome and describe an individual. Depending on each specific problem, a chromosome could be an array of real numbers, a binary string, a list of components in a data base, etc. (Andersson, 2001). A single individual represents a solution and a set of solutions forms a population. The fittest (archive) are then selected from the population for mating, where the combined genes result in a child. The children are reinserted into the population and the whole procedure begins again creating a new generation (Figure 2-2). The optimization process continues until the convergence of the population occurs, i.e. the best combination of genes has been reached or until the maximum number of generations has been reached.



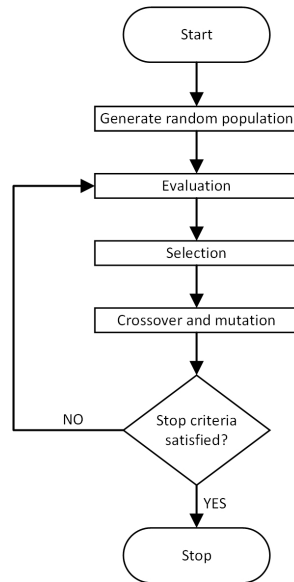


Figure 2-2. Flow chart for the GA process.

EAs have the tendency to converge towards a single optimized solution as the population becomes nearly identical early in the optimization process. Hence, niching and other techniques are used to create diversity within the population of feasible solutions. In MOO, niching pressure is applied to generate a diverse population along the Pareto front (Caldas, 2007). It is capable of identifying multiple optimal solutions within a single population. In MCDO, it is necessary to maintain diversity throughout the optimization process to generate new and improved trade-offs, which are ultimately the aim of the use of optimization in design.

The maximum performance of a design is the ultimate goal, yet, a design that performs simultaneously well on all objectives is rare, if not impossible. Conflicting objectives have an inverse relationship in terms of performance, where one objective can perform very well in a design but reduces the performance of another objective. It is the combining of multiple objectives in one design optimization problem that provides insight into each objective. It is vital to understand the trade-offs of the different performance metrics and the intricate relationships between the objectives governing the solutions. There is no single best solution in MOO but a population of potential ‘sub-optimal’ solutions, hence the notion of ‘optimality’ is different in this case.

Different methods, tools and algorithms can be used based on the aims of the designers. MOO methods can be split into two main approaches: scalarization and Pareto. The scalarization

approach is a method where a multi-objective problem is solved by translating it back to a single (or a series of) single objective scalar problems. In other words, multiple objectives are weighted based on the preferences of the designer *a priori* to form an aggregate function, i.e. before the optimization process runs. On the other hand, Pareto approach keeps the multiple objectives individualized throughout the optimization process (De Weck, 2004), which is discussed in detail in the next section.

#### 2.4.1. Pareto-Based Optimization

It is arguably stated that multi-objective thinking originated from economics with the definition of the ‘best decision’ being first considered by economics Professor Francis Edgeworth in 1881. The ‘best decision’ refers to decisions taken by buyers and sellers (micro-economics) or governments (macro-economics), which simultaneously optimize or balance several criteria. Multiple models such as taxation for example, applied multi-criteria thinking where ‘trade-offs’ had to be considered in order to optimize or balance economic decisions.

Vilfredo Pareto, a civil engineer, took up the study of philosophy and politics and was one of the first to analyze economic problems using mathematical tools. In 1893, Pareto became the Chair of Political Economy at the University of Lausanne in Switzerland, where he created his famous theory -- The Pareto Optimum, which he defines as: “The optimum allocation of the resources of a society is not attained so long as it is possible to make at least one individual better off in his own estimation while keeping others as well off as before in their own estimation” (De Weck, 2004). Widely accepted as the extension to Edgeworth’s work, it spurred the development of multi-objective methods in Applied Mathematics and Engineering forming the basis for Pareto-based efficiency and later optimization in the 20th Century.

Pareto optimization is “more realistic and useful for design”, according to Radford and Gero (1988), “because it allows subjective criteria to be taken into account.” Radford and Gero (1988) classified two general approaches for Pareto methods that can be applied in building design optimization problems: non-preference and preference.

A non-preference approach is limited to the production of information on non-dominated solutions (i.e. solutions that are not dominated by other solutions), it is therefore a bottom-up approach that provides information for the designer to make pertinent decisions. Those solutions

are referred to as Pareto front solutions, i.e. ‘best compromise’ solutions, as they always have a trade-off between each other that would be in favor of one objective over other objectives. The Pareto front is ultimately a representation of non-dominated, mathematically equivalent solutions (Gero & Radford, 1988; Caldas, 2007; Ciftcioglu & Bittermann, 2009; Benjamin, 2012). Non-dominated solutions can be graphically depicted by a two dimensional curve called the Pareto front (Figure 2-3).

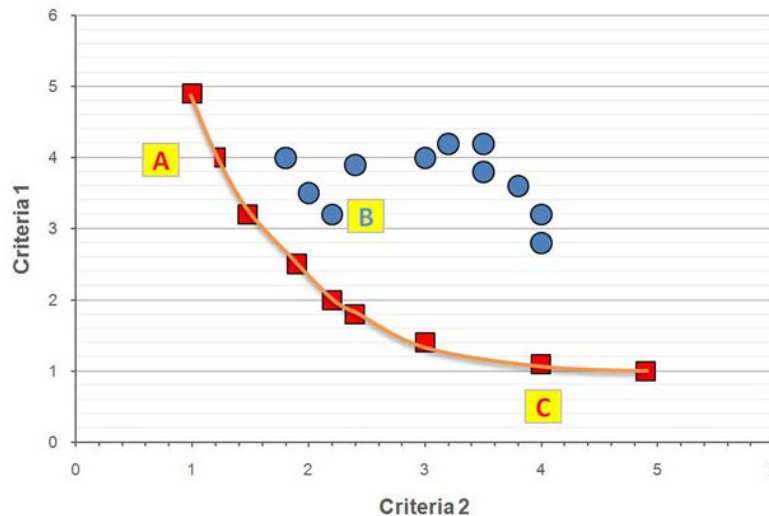


Figure 2-3. Pareto optimum (non-dominated solutions) formed by the red dots and dominated solutions formed by the blue dots.

A preference approach considers the designer’s trade-off preferences from the outset of the optimization process. It is therefore a top-down approach where the designer’s decisions are not based on the information provided by the optimization process but, based on the rational decisions about preferences of the designer. The set of feasible solutions is narrowed down and if possible, to identify directly a unique solution. Instead of computing the whole field of non-dominated solutions, this approach focuses on a subset or point that is of primary interest (Gero & Radford, 1988, p.217).

## 2.5. Precedents

This section reviews some of the practical research carried out on the application of optimization in design over the past 15 years. The case studies offer a broad review of the application of optimization from single objective optimization to multi-objective optimization design problems with their associated tools. Each case study will be briefly described in terms of the methodology and tools developed or used.

MOEAs have been used to aid design tools in increasing the number of discrete solutions generated for a design problem. The generated solutions are limited by hard constraints and are led by evaluation tools which in turn inform the generative tool of the performance of the design solution. The tools outlined below were mainly motivated by the challenge of enabling the designer to easily generate multiple solutions quickly to examine, explore and optimize for the ‘best’ design solution for a specific quantifiable design problem.

To engage the role of the designer in the optimization process, interactive MCDO tools provide a platform for the designer to interact and steer the trajectory of the design based on his/her design intent. It is with these interactive tools that the designer can participate in the optimization process, i.e. become integrated with the process. Several researchers have developed tools which require a Decision Maker (DM) to manipulate the process and hence the final outcome(s). The role of a designer is manifold and with the expanse of experience and knowledge that a designer develops undoubtedly ensures that a designer’s involvement is valuable. It is argued by DeLanda (2002) that the advent of digital design tools has downgraded the role of the designer to the “equivalent of a racehorse breeder”. On the other hand, interactive MOO tools do not reduce the designer’s role to a simple button press action or the feeding of data. The plethora of generated prescriptive information with the increased complexity of designs that surpass any person’s cognitive and computational abilities justifies the need and use of digital tools. It is the complementing of the computer’s strengths of fast calculations, endless memory, objectivity and extensibility with the designer’s experiences, knowledge and creativity that augments the design process and thus the final outcomes. Several precedents are discussed below that outline the benefits of using such tools.

The development of multiple integrated tools were inspired to amalgamate opposite poles of engineering and architectural design, for example Caldas (2001) developed a generative design system that uses evolutionary algorithms coupled with light and thermal performance evaluation software called DOE2.1E. The workflow of the system involves the use of Pareto-based evaluation of the results to determine trade-offs of the multiple solutions with the final decision-making left to the designer. The generated geometric solutions are based on the results of the trade-offs. The algorithm is able to determine which solutions are worth investigating and which are not based on their performance. The system evolved into *GENE\_ARCH* which expanded the

number of objectives that can be incorporated into the system beyond the light and thermal objectives (Caldas, 2007). The system is capable of generating complete building designs, both in terms of geometry, spatial layout and room characteristics (Figure 2-4) (Caldas, 2007).

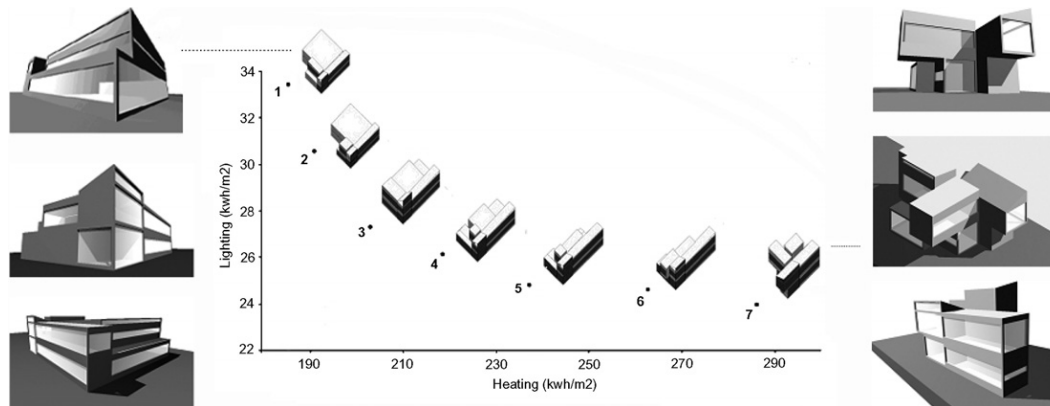


Figure 2-4. Pareto front for a case study generated by GENE\_ARCH, with best solution for heating (1) visualized on the left and best solution for lighting (7) on the right (Caldas, 2007).

Michalek & Papalambros (2002) developed a floor plan layout tool (Figure 2-5) that allows the designer to interactively manipulate the optimization process. Instead of automating the process completely, the interactive tool was designed to assist the designer with the design generation and evaluation process. The designer controls the process with the help of quantitative metrics and subjective judgments where appropriate, thus engaging in a creative exploration process of the design. During the optimization process, the designer can interact real-time, dynamically influencing the optimization process through adding, deleting and modifying objectives, constraints and structural units.

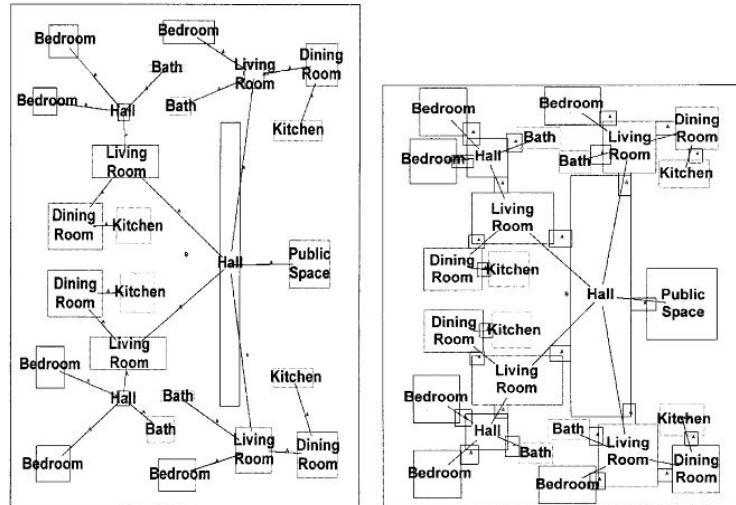


Figure 2-5. Iterations of the algorithm optimizing a sample apartment complex building with the objectives of minimizing annual cost and wasted space (Michalek & Papalambros, 2002).

The optimization engine defines the multi-objective problem as a single objective problem using an interactive technique called Interactive Weighted Tchebycheff (IWT). In this technique, the multi-objective problem is defined mathematically as a single function of the product of each objective by a linear weight and generally the objectives are non-competing. An optimization algorithm will then generate different points on the Pareto curve depending on how each objective is weighted. The designer's role is to dynamically fine tune the weights as the trade-offs are better understood. Michalek & Papalambros (2002) critique the process of defining the weights as being difficult and non-intuitive. The designer must generally see both some kind of form and performance metrics to comprehend design trade-offs.

Malkawi (2003) presented a performance-based design evolution system that allows efficient exploration of design alternatives. The workflow of the system is based on four main components: design evolution, performance evaluation, morph visualization and design evaluation. The design evolution component uses a GA based engine to generate the shape of the design instances. Following the generation of the design, a Computation Fluid Dynamics (CFD) analysis is automatically carried out to evaluate the thermal and ventilation efficiency and subsequently the metrics are fed back into the GA based engine. A continuous looping process runs until the best design solutions are reached.

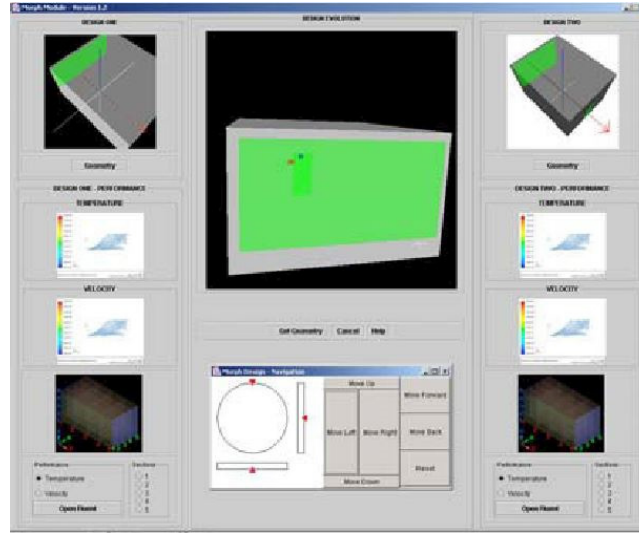


Figure 2-6. Morph-process GUI, one of the four components of system (Malkawi, 2003).

The morphing component records all solutions throughout the optimization run (Figure 2-6). This component allows the designer to view the solutions with their associated performance metrics, thus enhancing the designer's ability in exploring and discovering the 'best' solution. It also allows the designer to actively stop the optimization process and select an instance based on its form. The evaluation of this instance is performed automatically for the designer to visualize and check against the required performance criteria. Overall, the whole workflow creates discrete instances of designs that can be explored by the designer.

*eifForm* developed by Shea *et al.* (2003) combined structural evaluation with a generative design to generate complex structural geometry. Using structural shape annealing, which combines grammatical parametric shape generation, performance evaluation and stochastic optimization to support optimally directed exploration of discrete structural forms make up the generative method within *eifForm*. Structural shape annealing is based on Simulated Annealing (SA) which unlike GAs there is no crossover but only mutation (Kohonen, 1999). The choice of using SA does not guarantee that the exact mathematically optimal solutions are generated, but rather it supports optimally directed design exploration which supports the incorporation of structural grammar (Shea *et al.*, 2003).

According to Shea *et al.* (2003), this tool is capable of generating efficient and innovative planar trusses, single-layer trusses, and transmission towers as well as 3D trusses by further combining it with Custom Objects (Figure 2-7). (Custom Objects is a graph-based associative geometry

modeling system that combines geometric modeling and programming similar to what Grasshopper3D and Dynamo are today.)

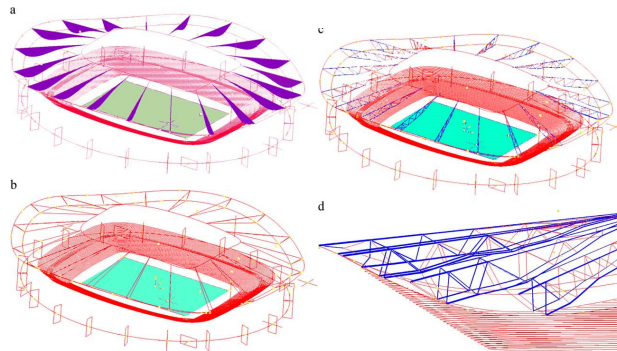


Figure 2-7. A set of optimized cantilever trusses with different unique spans generated by *eifForm* for a stadium roof model created in *Generative Components (GC)* (Shea et al., 2003).

The workflow requires the designer to input or create the initial geometric, generative and performance models that reflect the design intent. The maximum number of structural members that is generated can be defined in the generative model. Structural considerations such as the material type and loading are defined in the performance model. Overall, these parameters are representative of spatial and cost performance objectives established by the designer. *eifForm* is one the earliest performance-based generative design tools to combine multiple objectives as well as aesthetics via effective visualization and manipulation of the geometric and topological relations.

Keough & Benjamin (2010) tackled the use of MOO in design through the development of an automated workflow using Catia for parametric modeling, Autodesk Robot for structural analysis and modeFrontier as the MOO engine for the exploration of a wide design solution space and as an aid in decision making process. The workflow was applied on the design of a high structurally performing project called *Living Light*, a permanent pavilion in a public park in Seoul, South Korea (Figure 2-8). The optimization process would trigger Catia to feed its different parameter data into modeFrontier to carry out the MOO process. The geometry generated in Catia is then exported through a custom coded software called CatBot to Robot for structural analysis. The analysis results are then fed back into modeFrontier and exported as a text file.



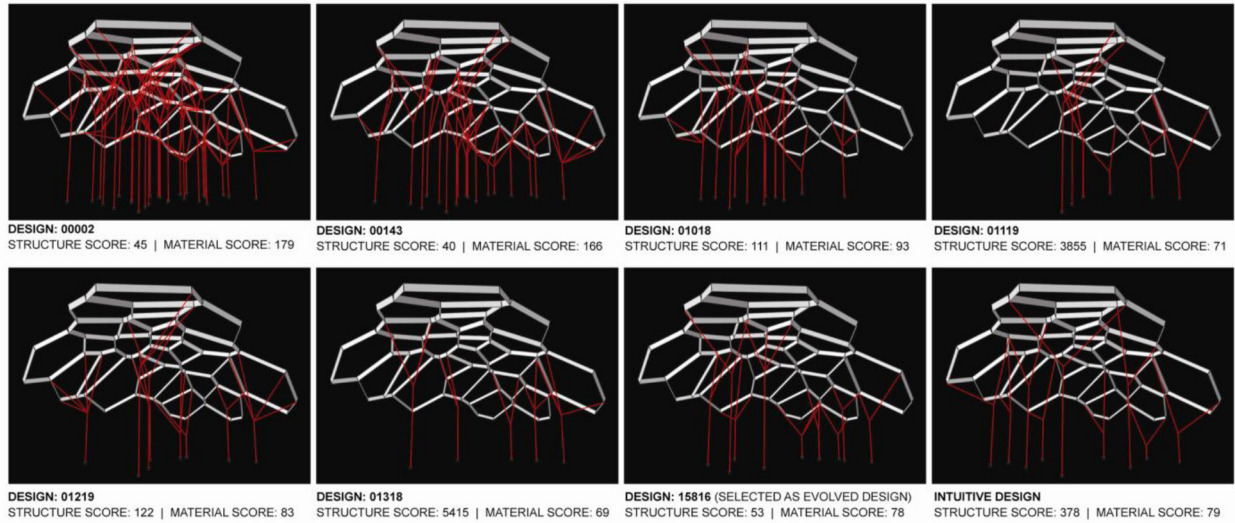


Figure 2-8. Several design iterations of the Living Light project (Koeugh & Benjamin, 2010).

*ParaGen* developed by Von Buelow (2011) is a more advanced method for interactive optimization. It has been under development for the past decade at the University of Michigan, Hydra Lab, however, it does not have a public release version. The advantage of *ParaGen* is the parametric design tool which gives it flexibility and adaptability for any design problem, in other words, it is similar to Grasshopper3D. The workflow is comprised of several tools: a GA engine that searches for well performing solutions, an associative parametric design software and, a performance evaluation software based on the aims of the design (Figure 2-9).

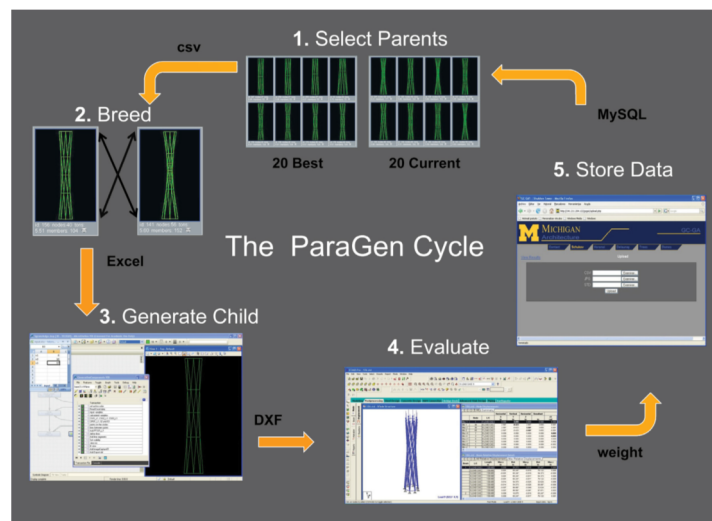


Figure 2-9. The five basic phases of the ParaGen cycle: select, breed, generate, evaluate, store data (Von Buelow, 2011).

Turin *et al.* (2011) applied *ParaGen* on two case studies: first case study used *ParaGen* to explore the morphology of a dome based on its structural performance and the second case study deals with the solar heat gain and daylight transmittance of a long span roof. The designer can interact on two levels in *ParaGen*: the generated solutions are stored in an online database where the designer can sort, filter and analyze solutions quantitatively in order to assess the solutions qualitatively and aesthetically. The designer can also interactively ‘mate’ preferred solutions to ‘breed’ new solutions based on designer’s aesthetic preferences.

*ParaGen* is not designed like other traditional MOO tools where the MOEA generates multiple solutions and sorts them automatically to form a Pareto front. Instead, it guides the designer towards the good performing solutions whilst giving the designer the freedom to manipulate the generation towards the sub-optimal solutions that better meet all the objectives, both measurable and non-measurable.

Cunha *et al.* (2011) combined the technology of MOEA with decision making technology to allow for a Decision Maker (DM) interaction. The interactive methodology developed by Cunha *et al.* (2011) is based on the use of MOO algorithms with the aim of exploiting and not necessarily exploring design solutions. Quantifiable objectives along with boundary conditions are initially defined by the DM. The algorithms then generate a large population of solutions with a wide diversity to create a Pareto front. Through multiple iterations and designer (DM) interaction, the optimization process is fine-tuned to suit the designer’s aesthetic preferences, by adding weights to the objectives (Figure 2-10).

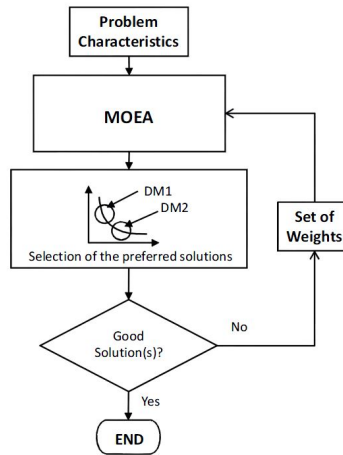


Figure 2-10. Workflow applying a priori use of weights to bias the solutions before the optimization process (Cunha et al., 2011).

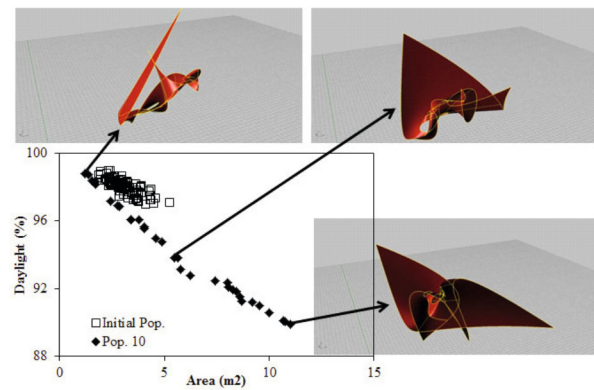


Figure 2-11. An example of one of the case studies explored by Cunha et al. (2011).

The method is designed to meet the quantitative and qualitative requirements of the early stages of design through combining multiple performance simulation tools with the MOO algorithms and a generative tool. The design process becomes quicker with greater functional and formal complexity with multiple evaluation tools integrated. The workflow aims to enable the designer to reach an optimal solution as several optimization runs can be done until the designer's objectives are satisfactorily met (Figure 2-11).

Gerber *et al.* (2012) developed H.D.S. Beagle tool which utilizes parametric design and MOO to influence design at its conceptual stages. The tool generates an array of design alternatives according to the designer's predefined parameter ranges and evaluates the energy performance of the solutions. Through the use of a GA, the system intelligently searches, ranks, selects and breeds the solutions (Figure 2-12). The tool was expanded to include cloud-based approach to

tackle higher degrees of geometric complexity and enable faster evaluation times (Lin & Gerber, 2013).

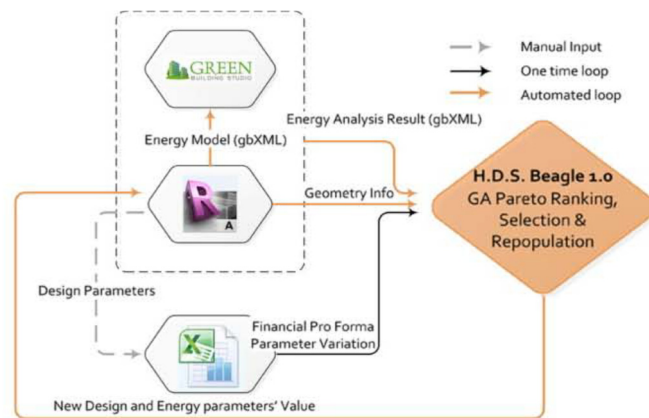


Figure 2-12. H.D.S. Beagle system architecture: Autodesk Revit is used for parametric generative purposes in combination with GBS for energy analysis and Microsoft Excel for the financial model (Gerber *et al.*, 2012).

H.D.S. Beagle utilizes Autodesk Revit for parametric generative purposes, Autodesk Green Building Studio (GBS) for energy analysis and Microsoft Excel for the financial model. The application of the tool was carried out on several hypothetical design scenarios (Figure 2-13) (Gerber *et al.*, 2012) and was used in a pedagogical experiment in order to verify reduced design latency and improved solution performance (Lin & Gerber, 2013).



Figure 2-13. An example of a sub-set of the solution space for the twisting tower scenario in which the multi-objective ranks the solutions and visualizes them for ease of manual decision making (Lin & Gerber, 2013).

### *2.5.1. Conclusion*

The notion of MCDO workflows and tools is not novel, yet the previous efforts presented all attempted at making MOO workflows more accessible to designers. The main aim of all these case studies was to improve the performance of designs through either ‘exploitation’ or ‘exploration’ or a combination of both. The common problems tackled by the researchers are the following:

1. Evaluation of subjective objectives such as aesthetics;
2. Real-time analysis (search; filter; sort);
3. Sensitivity analysis (qualitative analysis of resultant data and better understanding of the trade-offs);
4. Feedback loop: ability to make informed design decisions either within the optimization process or following the optimization process.

Not all previous efforts reviewed managed to address all these problems at once, however, the developed workflows and tools hold great promise for the field of MCDO and indicate a resonating effect of the increased use of such tools in design.

## **2.6. Summary**

In this chapter, a comprehensive overview of the application of optimization in general and Pareto optimization in particular has been provided. It is only in the past three decades that saw the notion of performance-based design reached the forefront of architectural research and discourse. MCDO has fascinated designers but as shown by the limited precedents reviewed, it has also been limited in its application due to architectural complexity and limited computing power. The impetus is to expand the application of MCDO to appropriate its potential of improving the performance of both the design process and the generated designs.

## Chapter 3: Creative Optimization Workflow/Tool

### 3.1. Introduction

Thomas Maver in his essay ‘Predicting the past, remembering the future’ made a wise prediction about the future of CAAD (Maver, 2002):

*“The more we know the more we can figure; the more we can figure the more we understand; the more we understand the more we can appraise; the more we can appraise the more we can decide; the more we can decide the more we can act; the more we can act the more we can shape; and the more we can shape, the better the chance that we can leave for future generations a truly sustainable built environment which is fit-for-purpose, cost-beneficial, environmentally friendly and culturally significant.”*

Thomas Maver was insinuating that the ‘more’ comprehensive prescriptive information a designer has, the ‘better’ a design will be. With existing technology, designers are already overwhelmed with data exchange and visualizations, and the various interfaces through which these interactions happen. The increased demand for integrating performance-based techniques into the early design stage requires improved exchange of data and visualization to contribute to faster design decisions.

This chapter describes an extensible, semi-automated workflow (‘Creative Optimization Workflow’) for the improved comprehension of MCDO solutions through the development of parametric visualization tools. The use of Grasshopper3D parametric definitions combined with Octopus, the MOO engine and multiple simulation tools bridge the gap between initially the quantitative and qualitative performance of a design, and secondly, between the overall design performance in the early stages of design. The proposed workflow with its associated tools expand the capabilities for MCDO support tools through improved design performance integration in multiple domains at the early stages of design, mitigation of interfacing and exchange issues, improved exploration of design solutions/options with better decision support aids and overall reduced design latency through the automation of the generation of a great number of solutions. The workflow and associated tools make a unique contribution as of currently there are no tools that exist in the Grasshopper3D ecosystem that aid designers in evaluating multiple solutions parametrically both visually and analytically.

### 3.2. Conceptual Idea

Kolarevic (2005) envisioned a performance driven generative tool that would produce a range of design solutions based on a set of multiple performance targets. The design solution space would then be evaluated through a performance feedback system based on both quantitative and qualitative performance criteria. Based on this abstracted version of the workflow, Grasshopper3D with its versatility was chosen for developing the workflow. Among its features is that it is an open-source software with an array of supporting plugins that range from simulation tools such as Honeybee and Ladybug for energy and daylight simulations to Karamba for structural analysis to data exchange tools such as Lunchbox for Microsoft Excel. An important tool that was vital for this thesis was Octopus, an EA-based open-source MOO engine that works natively in Grasshopper3D. It is one of the many useful tools that are accessible to the design community in the Grasshopper3D ecosystem but the only one that can carry out MOO.

With this in mind, other commercial software tools such as modeFrontier, ModelCenter and Optimus do not offer automatic linkage with Grasshopper3D unless the designer codes or programs a plugin that would allow for such a linkage. Another disadvantage of these robust tools is that they are not readily accessible for most designers. They have a steep learning curve and lack the great support community that exists for Grasshopper3D and its associated plugins such as Octopus. The major reason for not using commercial MOO software specifically for this thesis is because of David Benjamin's research involving the use of modeFrontier. Benjamin's (2012) research required the manual retrieval of 3D geometry from Rhinoceros3D and linking it with the quantitative metrics and the Pareto front in modeFrontier for a complete analysis of the solutions (Figure 3-1). For designers, the formal performance of a design is one of the essential elements of the overall performance of a design. In addition, the manual integration of the

multitude of data is tedious and subsequently hinders the detailed exploration of the design solution space unless the purpose of optimization is exploitation (Bradner *et al.*, 2014).

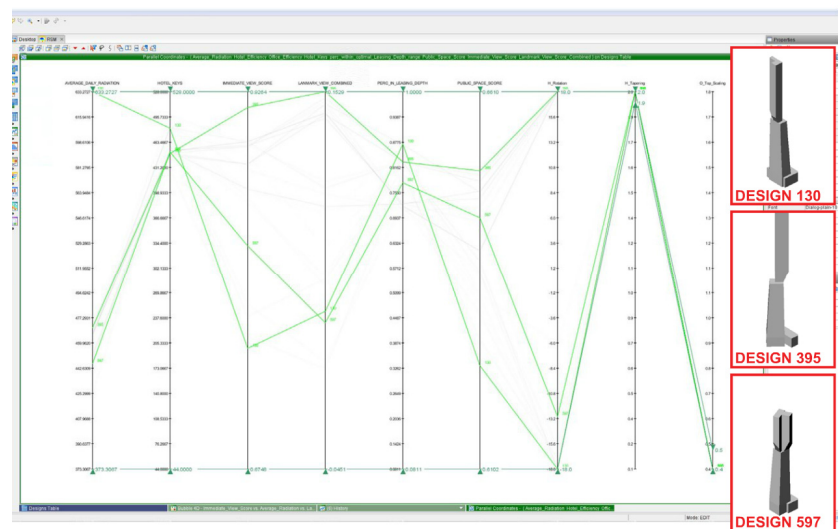


Figure 3-1. Filtering of solutions using modeFrontier and manual retrieval of the respective 3D solutions (Benjamin, 2012).

### 3.3. Creative Optimization Workflow Documentation

The workflow (Figure 3-2) includes the use of existing plugins for Rhinoceros3D: Grasshopper3D for parametric modeling, Octopus for MOO, DIVA for daylight factor analysis and Ladybug for views and glare analysis. Grasshopper for Rhinoceros was chosen as a platform for flexible, intuitive and integrated modeling (Vierlinger & Hofmann, 2013). The unique functionality of the semi-automated workflow is the ability to represent, manipulate, sort, filter and compare visually data of the design solutions using Grasshopper3D through the use of the ‘creative optimization tools’.

The design generation and performance evaluation is carried out in Grasshopper3D and using simulation software respectively. Solution quantitative data and 3D data is recorded and stored in Grasshopper3D; Octopus carries out Pareto ranking to identify potentially well-performing solutions. When the maximum generations are reached or when the designer stops the optimization run, the stored results can then be exported into a spreadsheet to be subsequently imported into Grasshopper3D. The ‘creative optimization tools’ can then be used to understand the solutions generated along with the 3D mesh solution. The results are displayed in Rhinoceros3D in a gallery format (radar charts, parallel coordinate plot graphs, 3D solution space graphs (Pareto graph), ‘form-based’ sorting and conditional domain searches).



Following this one-time loop, using the creative optimization tools, analysis can be conducted to explore the resultant solution space and fundamentally aid the designer in reaching a decision based on both quantitative and qualitative (formal qualities) analysis. Selection of the satisfactory/preferred solutions by the designer could be further optimized in Octopus by mating them to create more sub-optimal solutions or alternatively can be used for design development. The proposed tools are intended to be used by designers at the early stages of design, to integrate MOO with simulation and information visualization tools. The scalability of the tool is in its ability to be used for any design exploration whether skyscraper design or curtain wall optimization due to the flexibility of Grasshopper3D.

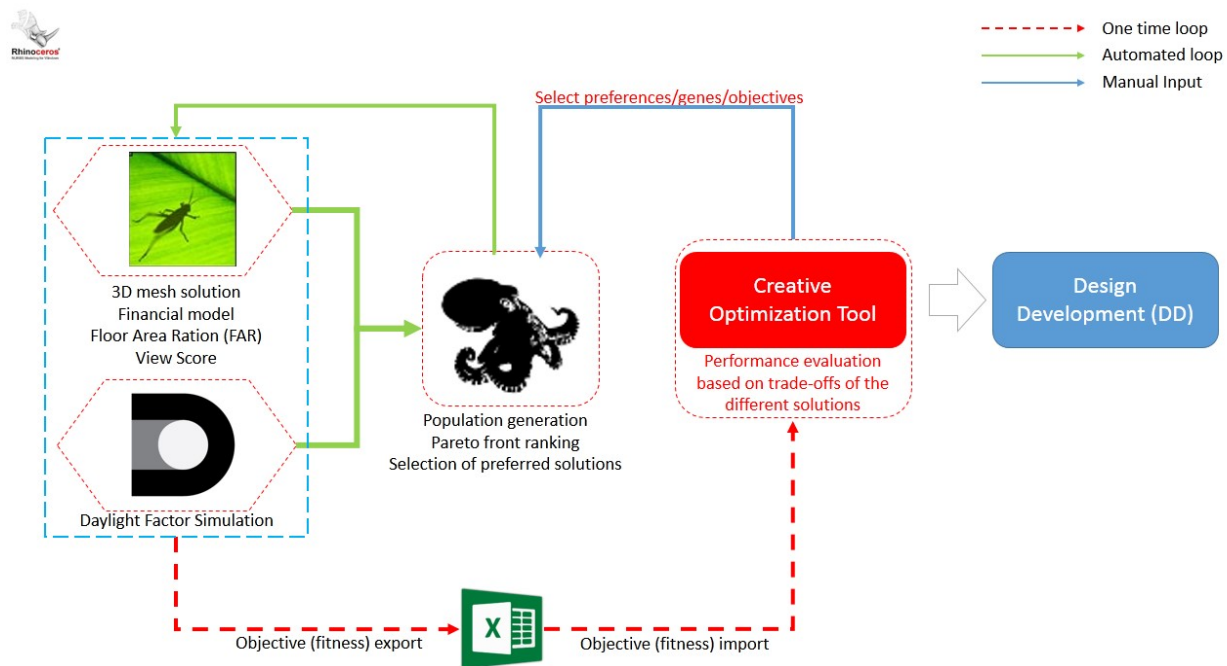


Figure 3-2. Creative optimization workflow.

Octopus is used as a black box engine to perform the optimization process in this thesis and thus the optimization routines were not altered. The thesis focuses on using a non-preference approach in Pareto-based optimization to generate the largest solution space possible. Preference methods, however, are feasible in the workflow and may allow the designer to explore through exploiting certain solutions. This can be done in Octopus during or after the optimization process.

### 3.3.1. *Parametric Model Setup*

Parametric design tools such as Grasshopper3D (GH) allow the designer to design with pre-defined constraints and parameters to quickly generate alternative design solutions through changes in the geometry. Hard constraints such as plot dimensions and height restrictions are defined in GH. Building information data (quantitative) such as program areas (office, hotel, residential & retail), profit calculations and geometry parameters are also defined in GH. Simulation plugins such as DIVA for daylight analysis can be used in the definition among other plugins based on the user's preferences and cognitive abilities. Octopus, the MOO engine, initiates the evolutionary run where each iteration (solution) is analyzed in terms of performance and the respective values.

Recorders in GH record both the 3D mesh of the solution and its respective performance values during the optimization process. Following the optimization run, all the relevant quantitative data is then streamed into a spreadsheet in Microsoft Excel, and subsequently imported back into GH using an Excel importer component (a plugin called Lunchbox). The solution space results can then be manipulated, sorted, filtered and analyzed via the 'creative optimization tools' (discussed later in this chapter).

### 3.3.2. *Octopus*

Octopus is an open-source plugin for applying EA to parametric design and problem solving. It has the capability to carry out an optimization search for multiple goals at once, producing a range of optimized trade-off solutions between the extremes of each goal. It is under the development by Robert Vierlinger in cooperation with Christoph Zimmer and Bollinger + Grohmann Engineers.

Octopus uses an iterative evolutionary search process; in each iteration a solution set is created. In GA terms, the solution set is called a generation which consists of a population of solutions that have different genes (design variables). The first solution set is generated randomly. The following generations are generated by an algorithm, emulating the biological reproduction by pairing solutions. After each iteration (solution), the genes of the two of the best performing solutions are combined to create a new set of genes. These genes form the design variables of the new generations (Vierlinger & Hofmann, 2013). This process mimics natural selection in a

limited sense; the fittest solutions survive, however, they are not necessarily the fittest solutions within the whole context. Solutions that fit the required objectives (fitness criteria) will be considered the highest performing designs. In order to prevent the algorithm from getting stuck at a local optima, a chance of mutation is defined in the algorithm, meaning that the generated population has a more diversified offspring (Vierlinger & Hofmann, 2013).

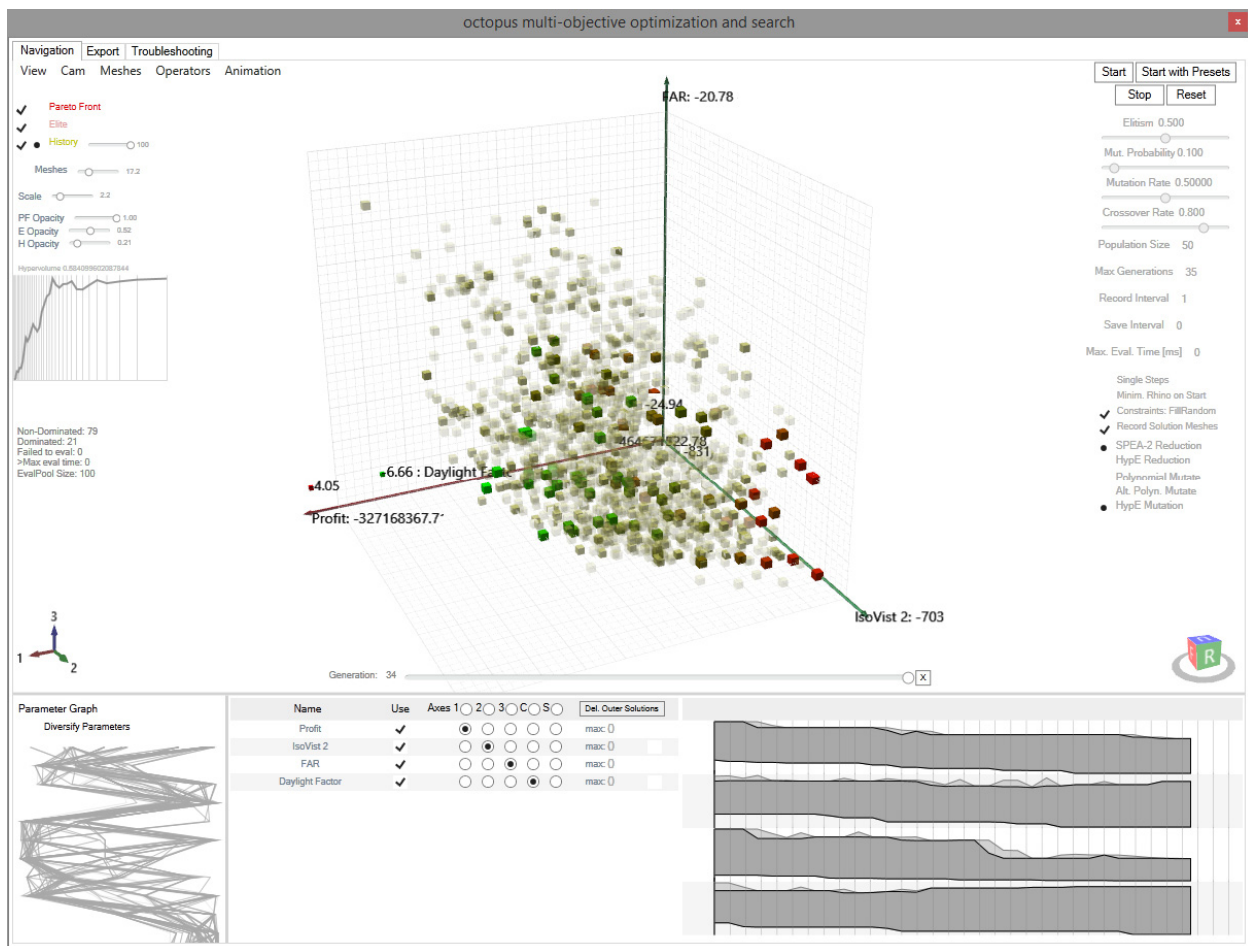
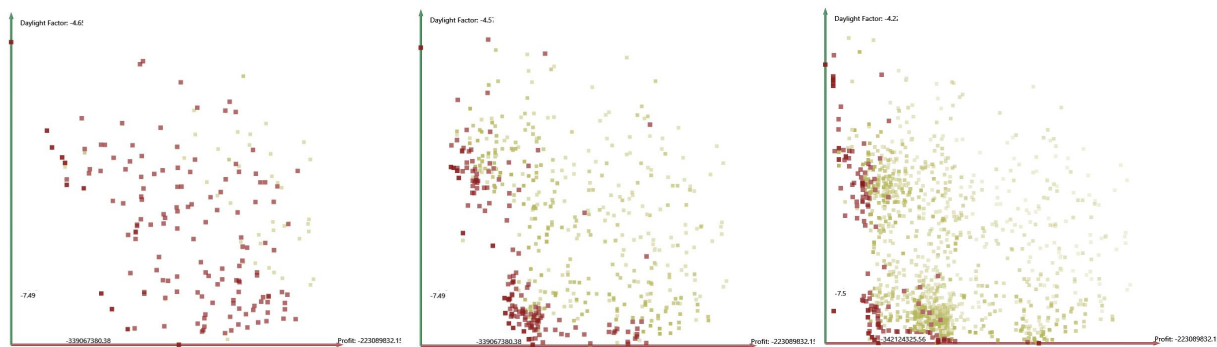


Figure 3-3. Octopus Interface.

During the optimization process, Octopus plots in real-time the results on a 3-dimensional graph, with each axis representing an objective; color is used for the fourth objective and size for the fifth objective. Octopus plots each solution in real-time as a cube (Figure 3-3). In other words, each cube is a mathematical representation of the performance and thus the rating of that particular solution. The solutions that form the Pareto front (non-dominated) are represented as red solution cubes. The best performing solutions are those nearest to the origin and the best performing solutions for a particular objective are those nearest to any axis. The light green

squares are generations from the archive which did survive over the generations (i.e. dominated) by other solutions (Figure 3-4). These solutions form the history of the optimization run. Instead of maximizing all objectives, Octopus minimizes the objectives; therefore the fitness values are adjusted accordingly in advance before the optimization process begins to achieve the required objective of either maximizing or minimizing (Vierlinger & Hofmann, 2013). All the solutions, including those beyond and on the Pareto front can be explored and analyzed through re-instating the solution via Octopus.



*Figure 3-4. The red dots slowly form the Pareto front formation over the generations (from left to right).*

Octopus offers an intuitive interface where the designer can zoom and explore the objective space, can filter solutions and add weights to solutions based on either their genotype (parameter) or phenotype (fitness). All solutions are stored in the history of the tool providing the flexibility to examine all design alternatives, redefine constraints and, verify and trace the continuity of the search process in a linear fashion. In addition, the history can be reused for modified designer preferences and initiate a new search process. These options encourage man-machine-dialogue and the incorporation of qualitative aspects of a design (Vierlinger and Hofmann, 2013).

Octopus is highly flexible with changing design problem definitions during or after the search process. Octopus offers two different reduction strategies (SPEA2 and HypE) and three different mutation strategies (Polynomial, Alternative Polynomial and HypE) based on the population size and number of objectives respectively. Elitism, mutation probability, mutation rate, crossover rate, population size and maximum generations can also be set by the designer manually before or during the optimization, to accommodate the designer's preferences and the scale of the design problem. These parameters are necessary to prevent or ideally mitigate the algorithm from

getting stuck in a deceptive optimization loop, i.e. convergence trap. The designer can choose the reduction and mutation strategy based on the design aim and with genotypic and phenotypic identification, ultimately the trajectory of the search can be altered in continuance with an existing population.

### *3.3.2.1. Search Algorithms*

The basis of Octopus search algorithms is a traditional GA which from the outset generates an initial population set using mutation to inform the following generations with their best properties (elitism). The GA engine uses a non-dominating sorting to find the Pareto front (Vierlinger & Hofmann, 2013). Octopus utilizes Strength Pareto Evolutionary Algorithm (SPEA-2) reduction in combination with the Hypervolume Estimation Algorithm (HypE) for multi-objective optimization algorithm for mutation (population method). The two algorithms are explained below outlining how they individually work, however, based on the designer's choice of the population and/or mutation strategy, they can be essentially combined to form a hybridized algorithm (Choudhary & Michalek, 2005).

#### Strength Pareto Evolutionary Algorithm (SPEA2)

SPEA2 is one of the most robust and efficient MOEA developed. The algorithm was first introduced by Zitzler *et al.* (2001). It is categorized as a non-dominant and archive based algorithm and was conceived with the notion of integrating different MOEAs together. As based on its categorization, the algorithm generates an archive (elite population) containing non-dominated solutions, i.e. Pareto front. With each new generation, the non-dominant individuals are copied to the external non-dominated set. SPEA2 has three main features:

1. Ranking value system: for each individual in this external set, a strength value is computed for a solution based on the number of solutions that dominate it and the number of solutions it is dominating.
2. Nearest neighbor density estimation technique which informs the algorithm of the density of solutions surrounding an individual solution, thus forcing a wider search area and prevents the algorithm from getting stuck in a local optima.
3. Enhanced archive truncation method guarantees the preservation of boundary solutions while being considerate of memory consumption during the optimization process.

Overall, SPEA2 has outperformed other alternative MOEAs in terms of convergence, diversity and computing resources (Zitzler *et al.*, 2001; Abraham *et al.*, 2005). According to Vierlinger & Hofmann (2013), SPEA2 lacks a robust strategy to ensure the widest possible extent of the solution set can be explored. The extent is defined by the truncation of the archive (Pareto front) which is essential to ensure the survival of extreme solutions. Therefore, extreme solutions before truncation may be not considered as they are eliminated by elitism and so, a lack of diversity in the solutions may exist depending on the breadth of the design parameters.

#### Hypervolume Estimation Technique (HypE)

Hypervolume Estimation Technique (HypE) is an indicator-based algorithm, where the MOO problem is condensed mathematically into a single-objective problem. The optimization is then carried out on the indicator value itself instead of optimizing all multiple objectives directly. A Hypervolume Indicator (HI) is a numerical value that represents the measurement of the area or volume of the non-dominating Pareto front. The aim of the algorithm is to maximize its value as much as possible and with the best case scenario yielding a set of Pareto-optimal solutions that have the following properties (Bader, 2009; Vierlinger and Hofmann, 2013):

1. They are near to the real Pareto front;
2. They are diverse and evenly distributed;
3. Many extreme solutions are included.

#### *3.3.2.2.Filter strategies*

Octopus provides graphical aids (filter strategies) for the designer to understand the optimization process as well as help guide the designer to make optimization decisions. In addition to the graphical aids, the interactive objective space with its associated filter buttons allow the designer to apply filter strategies and, identify sensitivity relations between parameters and objective or phenotypic characteristics (Vierlinger and Hofmann, 2013).

The convergence of solutions is a necessary method of verifying whether the search method has explored all possible solutions or not. Depending on how “conflicting” the objectives are and how wide the parameter ranges are, the more time Octopus will take to reach a convergence. As soon as convergence occurs and the designer is satisfied with the solution space, the search can be stopped. Based on the scale of the design problem and the designer’s trial and error, the

number of generations can be determined from the outset. However, if the solutions fail to converge, the number of generations can be increased and the search process can continue. It has to be stated though that an optimization algorithm is NOT a replacement of good thinking and thus there is no universal answer as to how many generations are needed to solve a design problem. Octopus provides the ability for the designer to view the history of the solutions and ideally identify if indeed solutions have converged or if the algorithm is stuck in a deceptive trap (for example, a local optima).

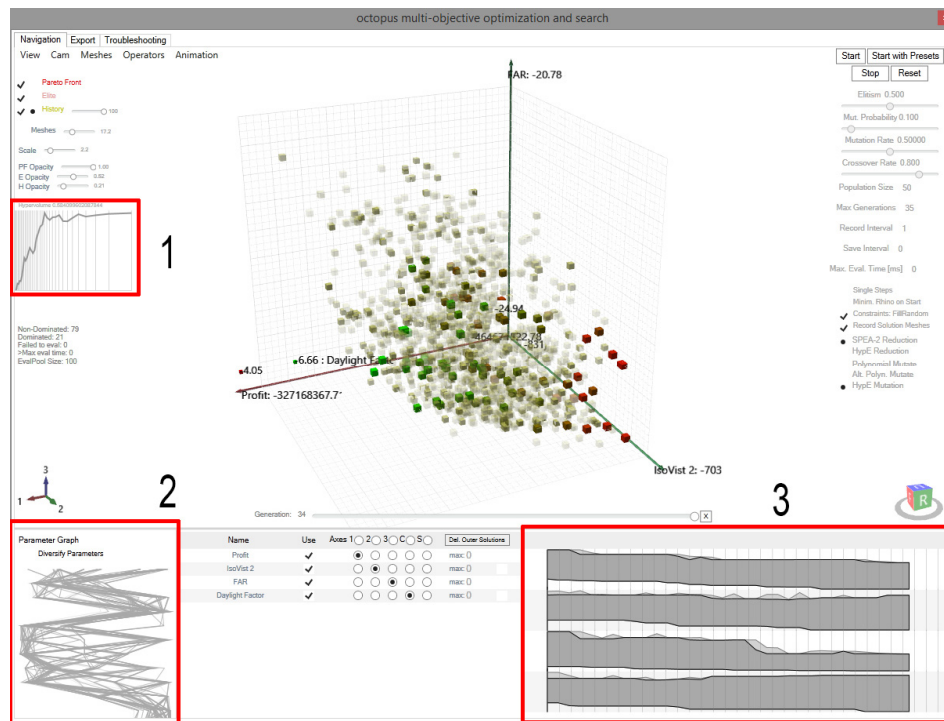


Figure 3-5. Octopus interface: (3-5.1): Hypervolume graph; (3-5.2): Parameter graph; (3-5.3): Convergence graph.

Three graphs aid the designer in identifying convergence of solutions: Hypervolume graph (Figure 3-5.1) is a history graph representation of the mathematical measure for the spread of solutions: when the line begins to flatten out, the solutions are converging. The domain of the graph is normalized from the 0 to 1, with a maximum spread of solutions at 1. Parameter/genetic distance graph (Figure 3-5.2) is similar to the genome graph in Galapagos (a single-objective engine native to GH), where each polyline represents a solution's parameter values; and as the solutions converge, the polylines begin to re-iterate closer to each other (i.e. become denser). Each polyline is drawn through comparison of the different configurations by a single accumulated fitness value. In Figure 3-5.3, each convergence graph represents one objective

scaled to a domain of 0 to 1; light grey represents the elite solution domain and the dark grey represents the Pareto front domain and is updated for each generation. A completely dark grey graph means that the algorithm has found that all elite solutions are also the Pareto-non-dominant solutions as well. The changes in the graph throughout the generations is a representation of the truncation of the Pareto-front results, which is done by the reduction strategies (Vierlinger & Hofmann, 2013).

### 3.3.2.3. Preference search

Octopus offers the ability to configure the environment of the GA through data input, filtering abilities (Boolean functions) and, parameter and objective preferences. The evolution and search process is open-ended but in Octopus, each solution can be considered a solution to further develop.

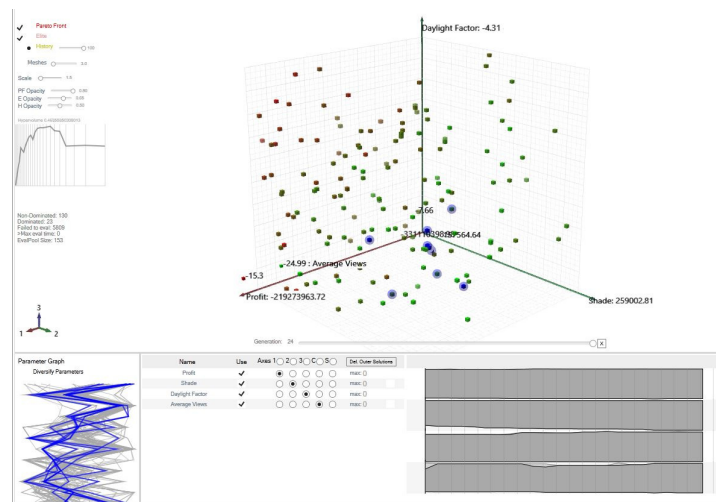


Figure 3-6. Preferred solutions selected (solutions highlighted in blue).

A technique known as ‘attractor points’ can be used in the objective space for the expression of user preferences to direct the search (Figure 3-6). As soon as convergence of the solutions slowly is formed and with the designer aware of the relationships and trade-offs, the designer can include his/her preferences to the process. The attractor points chosen will initially truncate Pareto fronts and will refine the search to be around those preferred solutions. This level of interaction in a GA-based tool is analogous to the design problems and proves to be successful (Vierlinger & Hofmann, 2013).



### 3.3.3. *Simulation Tools*

Two simulation plugins were used as part of the case studies discussed in Chapter 4. The affordance offered by these tools is significant as they are originally based on open source, verified algorithms, thus making them open source as well. Each tool is discussed in detail below with regards to the systems and algorithms that are incorporated in each tool.

DIVA (<http://www.diva-for-rhino.com>) is a GH plugin that can be directly run from the Grasshopper3D interface using a pre-built definition provided by Harvard GSD (SD)<sup>2</sup>. This plugin component runs Daysim, the calculation interface, which employs a commonly-used calculation engine, Radiance.

Ladybug is an open source GH plugin that helps inform designers about environmental performances of their designs. The components specifically used for the case studies are the ‘views’ and ‘glare’ components. The ‘views’ component evaluates the percentage of the available views for the whole building geometry in regards to certain visual points and the immediate surroundings. The glare component bounces direct sunlight vectors off the geometry under evaluation. The geometry and its surroundings are assumed to be specular. The number of points that strike the surrounding public spaces (streets and outdoor spaces) are computed. Both DIVA and Ladybug use EnergyPlus Weather files (.EPW) which are loaded into the definitions from the outset based on the location of the building under evaluation. Both tools are available for free as part of the array of free plugins available for Grasshopper3D.

### 3.3.4. *Representation & Manipulation of results (Creative Optimization Tools)*

Octopus is limited to visualizing multidimensional data in a 3D graph, where the Pareto front is visualized. To overcome this limitation, there was a need for the visualization of the significant amounts of data in a meaningful way to make MCDO solutions more accessible for the designer. Both quantitative data and formal qualities of the designs had to be considered as part of the visualization tools to enable a more holistic approach to MOO based design.

The use of various visualization techniques such as parallel-plot coordinate graph in multidimensional problems is not novel; several MOO software have the ability to visualize data produced in various and combined ways. Visualization uses the “high-bandwidth human perceptual and cognitive capabilities to detect patterns and draw inferences from visual form”

(Andrew, 2005). Therefore, visualization enables the designer to quickly identify and contextualize certain trends and in doing so reacting to such information with decisions and projections.

The developed visualization tools in this thesis have a dual role in both visualizing quantitative multidimensional data in a meaningful manner and in visualizing the 3D solution meshes. The choice of the tools to be developed was based on evaluating the needs of the workflow, i.e. what tools are needed to enable the designer to explore a large set of solutions in an informative holistic manner? In addition, the tools are interactive to enable the designer to search, filter, sort, and compare solutions. The user-engagement is fundamental in synthesizing, evaluating, presenting and qualitatively assessing all solutions or subsets of it. It also aids the designer in determining the hierarchies of the data as well as its influence on the design.

Since the formal qualities cannot be mathematically defined and is a subjective objective, the Decision Maker (designer) has to determine which solutions are aesthetically pleasing and which are not. There were several attempts at developing tools that enable the designer to determine which solutions to further optimize or to take further into the design development stage (discussed in Chapter 2). Yet, the tools developed as part of this thesis are unique in addressing a gap that currently exists in the visual representation of MOO based results in Grasshopper3D.

Four tools were developed as part of this thesis (Appendix #1): Parallel-plot coordinate graph, radar-based chart, form-based choice, and Pareto graph. These tools can be used on their own or in combination to filter, sort, compare and choose preferred/satisfactory solutions. They can be used with any other Grasshopper3D components and are dependent on the recorded data (3D meshes with their associated quantitative performance values) during the optimization process. The tools enable the designer to visually compare solutions in a gallery alongside quantitative values of the respective solutions. The choice of gradient colors of red (worst performer) to blue (best performer) is based on the standard used in most performance simulation software (Figure 3-7). Each tool will be explained individually as to how it works and what it visualizes.

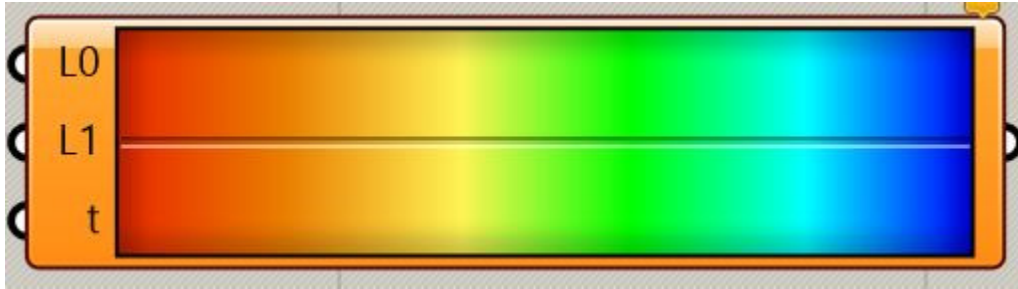


Figure 3-7. Color gradient used to visualize the level of the performance of the sorted solutions (red: worst performer; blue: best performer).

### Parallel-plot coordinate graph

The parallel-plot coordinate graph is a standard method of displaying multivariate results in a chart. Each individual objective is represented on a vertical axis and the values of each objective are normalized to fit on the length of each axis which are then displayed in Rhinoceros3D. Accordingly, each individual solution is then represented by a polyline across the multiple objectives. Conditional domain (subset) searches (in terms of either objective ranges or percentiles) of any objective can be conducted to filter solutions based on the user's preferences and aesthetics sensibilities (Figure 3-8). Subsets of subsets can also be carried out to refine the search domain. Using subset searches, the designer can filter design alternatives from different perspectives; for example, from an owner's perspective the main concerns would be profit generated and an iconic design. Other perspectives can be explored such as from the architect's or from the public's perspective, etc.

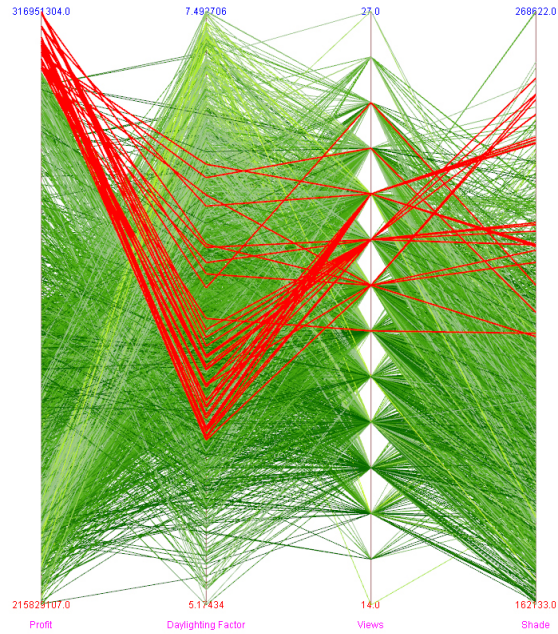


Figure 3-8. Subset searches are represented as red polylines.

During either individual solution searches or subset searches, the 3D geometry is displayed in Rhinoceros3D to visually compare the formal qualities of the solutions and to establish a link between the performance metrics and form. Subset searches can be coupled with radar-based chart and/or the parallel coordinate plot graph based on the designer's needs (Figure 3-9). All the solutions, including those on the Pareto front can also be explored using the parallel-plot graph as well (Figure 3-10).

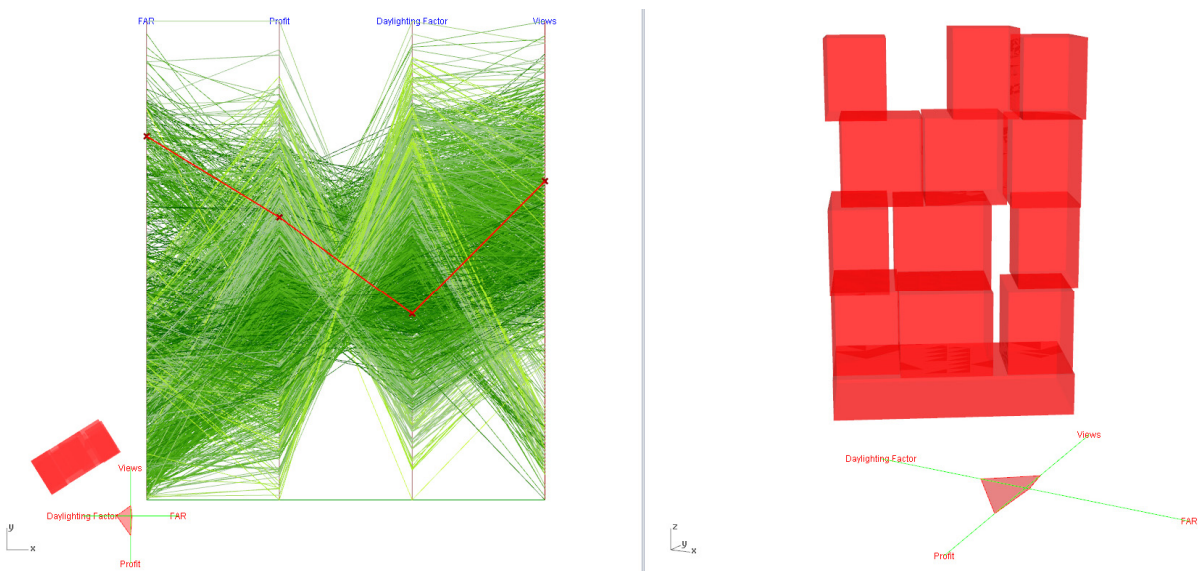


Figure 3-9. Individual solution search using parallel plot graph and radar chart.

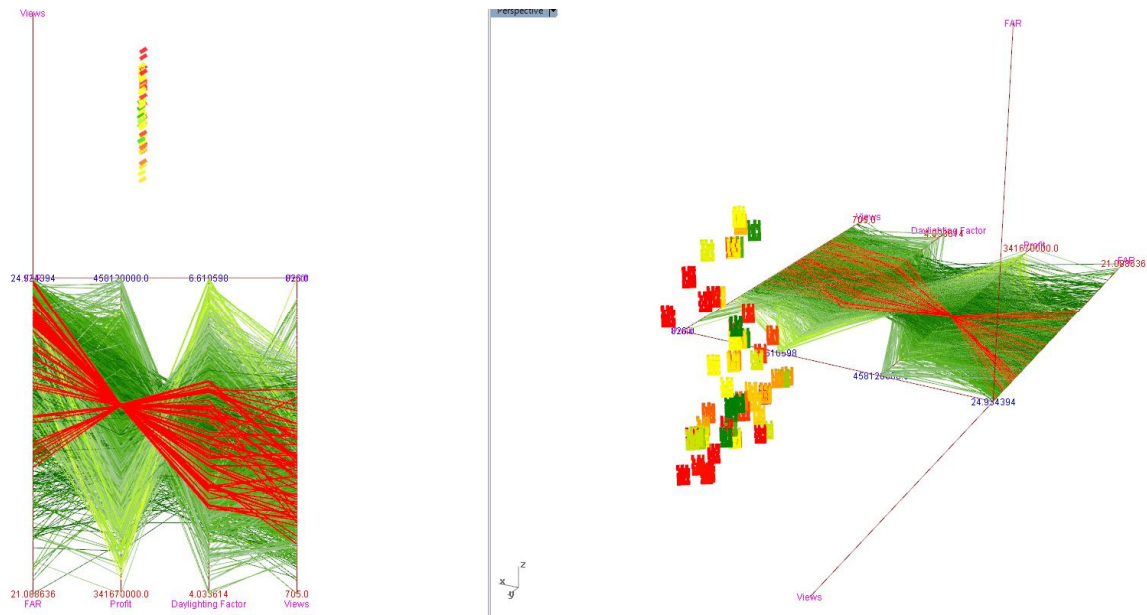


Figure 3-10. Subset search using the parallel plot graph and the Pareto graph.

### Radar-based chart

The radar-based chart (Figure 3-11) is another standard method of displaying multivariate data in a chart. The number of axes equals the number of objectives and the upper and lower bounds are normalized according to the length of the axes. The objectives under evaluation can be manipulated by the designer to examine certain objectives instead of all of them at once. The chart forms a polygon of the performance of a solution (the larger the area of the polygon, the better performing a solution is). The chart is very useful when comparing a small number of solutions to narrow down choices or to gain a better understand of trends in geometry within a subset. Overall, by simply using a number slider to select a solution, solutions can be compared using a radar-based chart and a parallel coordinate plot graph to identify trade-offs of each solution.

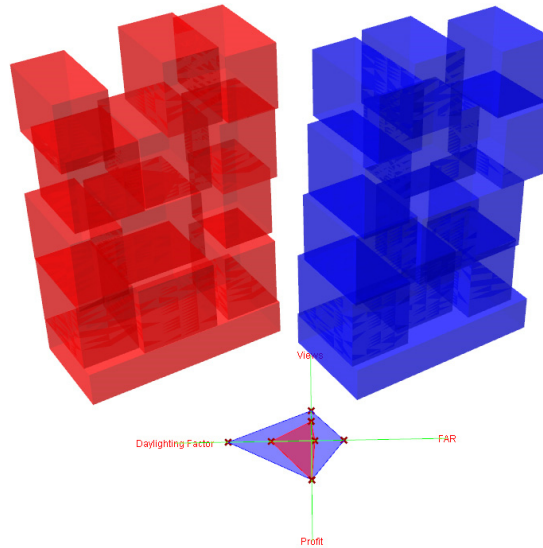


Figure 3-11. Comparison of two solutions using the radar-based chart tool.

### Form-based choice

Using an additional plugin called ‘Human’ developed by Andrew Heuman, a checklist component is added to select individual solutions based on solely the formal qualities from within a certain subset of solutions. The solutions can be further compared and sorted based on a specific objective or on all objectives. This tool is particularly useful in aiding the designer to narrow down potential satisfactory solutions through evaluating the formal qualities of the solutions as an explicit objective (Figure 3-12). The selected solutions are displayed in a gallery to visually compare solutions and thus evaluate more critically.

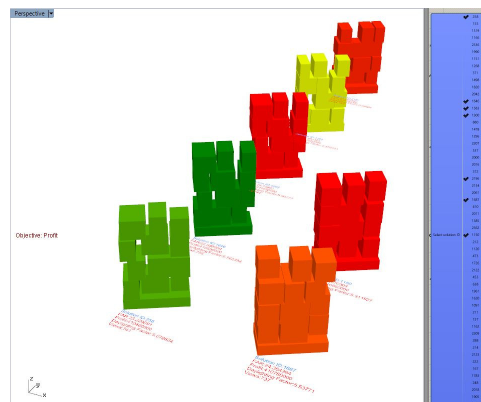


Figure 3-12. Selecting the solutions from a subset search based on formal qualities alone.

The Pareto graph is rebuilt based on the recorded data from the optimization run (Figure 3-13). It is the exact replica of the Pareto graph in Octopus but with a greater ability to filter solutions based on a subset search or to locate individual solutions within the plethora of solutions in the graph (Figure 24). In addition, the 3D geometry of the results are displayed within the graph to quickly evaluate the trade-offs of the solutions.



These tools are developed to encourage ‘creative optimization’ where the designer can explore not only the Pareto front results but all the solutions generated. Preferred solutions can then be selected in Octopus to run a preference-based optimization run where the genetic setup of the chosen solutions is ‘mated’. The ability to explore solutions is necessary to gain a complete understanding of the objectives and improved understanding of the designer’s preferences. It is through the juxtaposition of extreme solutions, that a platform is created where serendipity in a design can happen. The tools are developed in GH to streamline the process and to reduce the difficulties in dealing with multiple interfaces and/or data exchange issues. In addition to the graphical aids in Octopus, the designer can determine the optimization trajectory, identify

bottlenecks, enable rapid design exploration with evaluation of formal qualities as an implicit objective, quick visualization of the cause and effect of the quantified trade-offs and overall reduced design latency given the automation of the generation and evaluation of the generated design solutions.



## **Chapter 4: Case Studies**

### **4.1. Overview**

In this chapter, two retrospective designs are considered as case studies to evaluate initially the workflow and its tools, and secondly to critically explore the application of MOO in the early stages of design. High density buildings are chosen as they are seen as one approach of dealing with the current built environmental issues. High-density buildings (skyscrapers) need to perform on multiple fronts for them to become more viable and arguably more attractive to both developers and occupants

Two case studies were chosen: the De Rotterdam Tower designed by OMA built in Rotterdam, Netherlands and the second case study is the Bow Tower designed by Foster and Partners built in Calgary, Canada. These two buildings were chosen to test the flexibility of the workflow as they are characterized by different formal languages as well as different site and program requirements. It was also necessary to contextualize the results and test whether there would be any improvements in the formal qualities of the designs and in the quantitative performance compared to the original designs. Both buildings were designed within multiple constraints that exist natively through the respective building bylaws of the cities that they were built in such as Floor Area Ratio (FAR) restrictions and/or building height restrictions. All these hard constraints are considered for each design as part of the parametric definitions.

Each case study is discussed in detail in terms of which objectives are evaluated and why. A discussion of the results is carried out for each case study to evaluate the impact of the developed tools on the workflow, as well as a quantitative and qualitative analysis of the results followed by conclusions drawn from the overall study.

### **4.2. Case Study 1: De Rotterdam**

The ‘De Rotterdam’ Tower (Figure 4-1) designed by OMA (Rem Koolhaas) is used as a test case to design and evaluate the effect of the workflow. De Rotterdam, located in Rotterdam, Netherlands, is described as a ‘vertical city’ by Rem Koolhaas; it is a mixed-use building with the aim of maximizing efficiency and profit. De Rotterdam Tower consists of three stacked and interconnected towers via a plinth with a total area of 160, 000 m<sup>2</sup> and a total height of 150m (ArchDaily, 2014).



*Figure 4-1. De Rotterdam Tower.*

A replica of the De Rotterdam Tower (Figure 4-2) was parametrically modeled using GH (Appendix B) with the exact realistic plot dimensions, building height, number of floors, orientation of the building and the respective areas of the four main program functions: retail, office, residential and hotel.

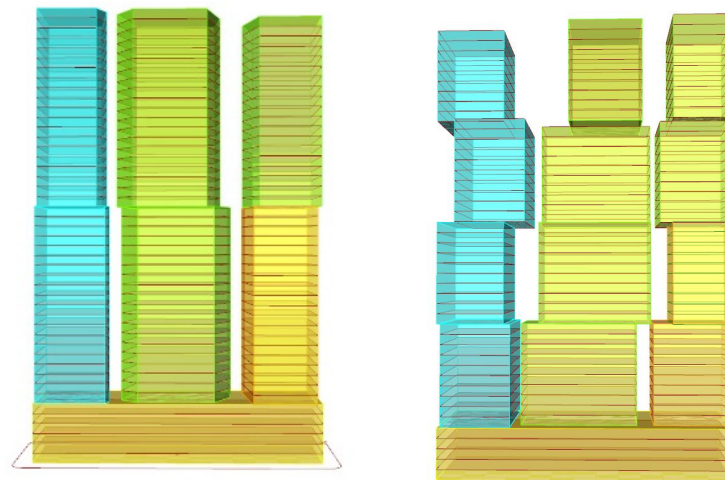


*Figure 4-2. De Rotterdam parametric replica.*

Two optimization runs were conducted using four objectives: maximizing the Floor Area Ratio (FAR), and thus financial profit, average daylight factor and views. These parameters were chosen because they are relevant to many mixed-use buildings, and easy to evaluate at this stage of the design where the design lacks geometric complexity. Bias in the objectives was not included in the optimization from the outset to expand the range of possible solutions and to

encourage exploration of the solution space. In addition, the same optimization settings such as the mutation rate were used for both De Rotterdam optimization runs to ensure consistency.

The first optimization run involved beveling of the individual towers (Figure 4-3) in the belief that it would increase views and subsequently natural daylight; however, significant area was lost and thus profit. Other issues with this run were the resulting form, the beveling based on the authors' opinions did not yield any improvements in the formal qualities of the design. The poor form performance resulted in exploring an alternative parametric definition and thus an alternative formal language. The second optimization run divided the individual towers into four stacked blocks (Figure 4-3) to create a more dynamic looking building and as a way of mitigating loss of floor area resulting from the first optimization run.



*Figure 4-3. Left: De Rotterdam bevel iteration; Right: De Rotterdam stacked block iteration.*

#### *4.2.1. Model overview: the design loop*

The design generation is carried out by a parametric model (Figure 4-4) where the solutions are evaluated by a Boolean function to ensure that the generated model is within the site and height constraints. If the model does not satisfy the Boolean function, it is discarded and a new set of design variables are generated. Once the design satisfies the condition, a performance evaluation of the solution is carried out. The respective quantitative data and 3D mesh solution are then stored. The amount of generated solutions is determined by Octopus and if the maximum amount of generations (and solutions) have been generated, the optimization process stops.

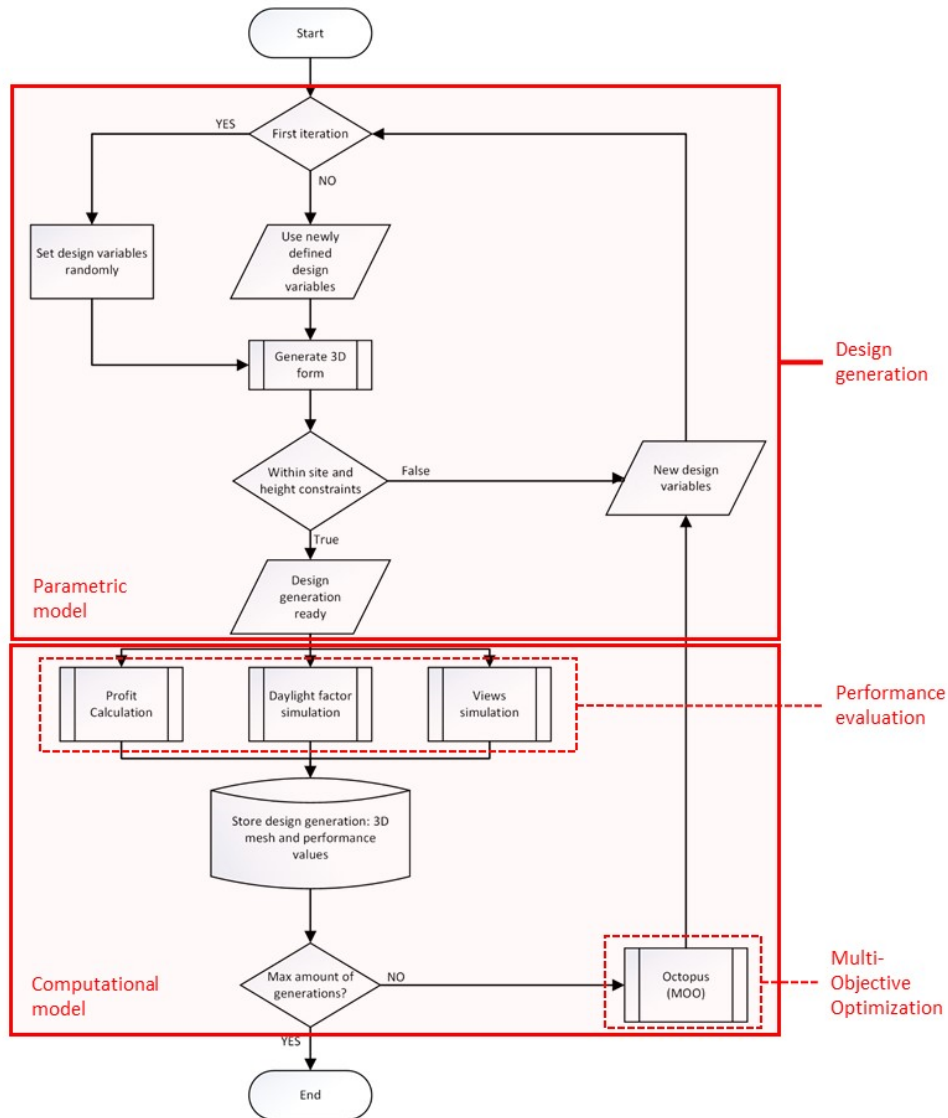


Figure 4-4. Generative model components of De Rotterdam Tower optimization runs.

#### 4.2.2. The parametric model

The parametric definition of the geometry (Appendix B) involved limiting the number of floors per individual tower to 37 floors in which the bottom block would be limited to a maximum of 19 floors and the top block to 18 floors. The retail plinth was limited to 7 floors giving a total of 44 floors as originally designed and constructed. The number of floors remained the same for both optimization runs.

The beveling of the towers is carried out using an ‘evaluate curve’ component which evaluates a point along the curve to which a new vertex is created and thus a new polygon is formed. The

newly formed polygon is then extruded to form the 3D form. For the stacked blocks, each tower was divided into four blocks, three blocks of 9 floors each and the last block of 10 floors giving a total of 37 floors per individual tower. The parametric definition of both runs also ensures that there is always a core zone connecting all vertical blocks together occupying an area of 30% of the individual floor area. The lateral movement of the individual blocks is limited to a XY domain to prevent overlapping of the towers with each other and to ensure it is within the building plot limits.

The individual program areas are divided by the individual floor area of each block in order to generate an integer value for the number of floors for that block. The range for each individual program area is  $\pm 10\%$  of the original program areas.

The Floor Area Ratio (FAR) is defined as the gross floor area divided by the plot area. The higher the FAR, the more square footage there is in a building for a given site.

The financial model is based on the difference between the construction costs and selling prices per square footage in Rotterdam (EC Harris, 2006; DTZ, 2014; 44 floors, 2015) for each program function, for example, retail and hotel have different construction costs and subsequently different selling prices (Appendix B). The financial model is adaptable and can be made more comprehensive by including expected operation and maintenance costs. The financial equation used is as follows:

$$\begin{aligned} \text{Profit generated} = & (\text{Retail area} * (\$800 - 250)) + (\text{Office area} * (\$350 - 250)) + (\text{Residential area} * (\$464 - 204)) \\ & + (\text{Hotel area} * (\$600 - 354)) \end{aligned}$$

The views score is evaluated using a component called ‘IsoVist’ in GH which projects points radially, with those not striking nearby geometry returning a ‘false’ Boolean value; their count provides the “views” score (the higher the value, the better the score). The number of points projected are equal for both optimization runs to maintain consistency.

Daylight factor is defined as the ratio of internal light level to external light level measured in illuminance at the working plane level of 0.762m above ground level. The analysis planes are modeled accordingly at working plane level and modeled on random floors across all the individual towers. The material chosen for the exterior glazing is ‘double pane – lowE’ with a 65% transmission, generic floor with a 20% reflectance and high reflectance ceiling with a 90%

reflectance. All default radiance parameters such as the number of bounces (-ab) are used. The final daylight factor percentage is the average of all these values for the building.

#### *4.2.3. Analysis of results*

##### First optimization run (beveled towers)

The first run involved the beveling of the individual towers (Figure 4-3) with the hypothesis of increasing the daylight factor, views score and to create a more ‘aesthetically’ pleasing form(s). This in turn would translate to increased revenue from the increased views score and also the improved daylight performance would translate into less artificial light, hence less energy use (implicit relationship) (Koster, 2004).

Sensitivity analysis of the solutions show that some of the highest FAR figures resulted in the highest daylight factor, but lower profits and average views score (Figure 4-5). The reason for this relationship is as a result of the beveling, square footage is lost leading to lower profit but higher daylight factor. Therefore, the beveling did result in more area being exposed to daylight thus leading to higher overall values. The hypothesis proved somewhat correct, that the beveling would lead to higher daylight factor and through subjective inference, this will lead to a mediocre increase in property retail values for the residential zones which may ideally mitigate losses in profit. Through further sensitivity analysis, some of the highest FAR also resulted in a very high profit and views score, but significantly low average daylight factor (Figure 4-6). This is due to ‘less beveled’ solutions which led to more square footage, thus directly increasing profit. The formal quality of the design solutions, however, is not particularly better according to the author’s opinions.

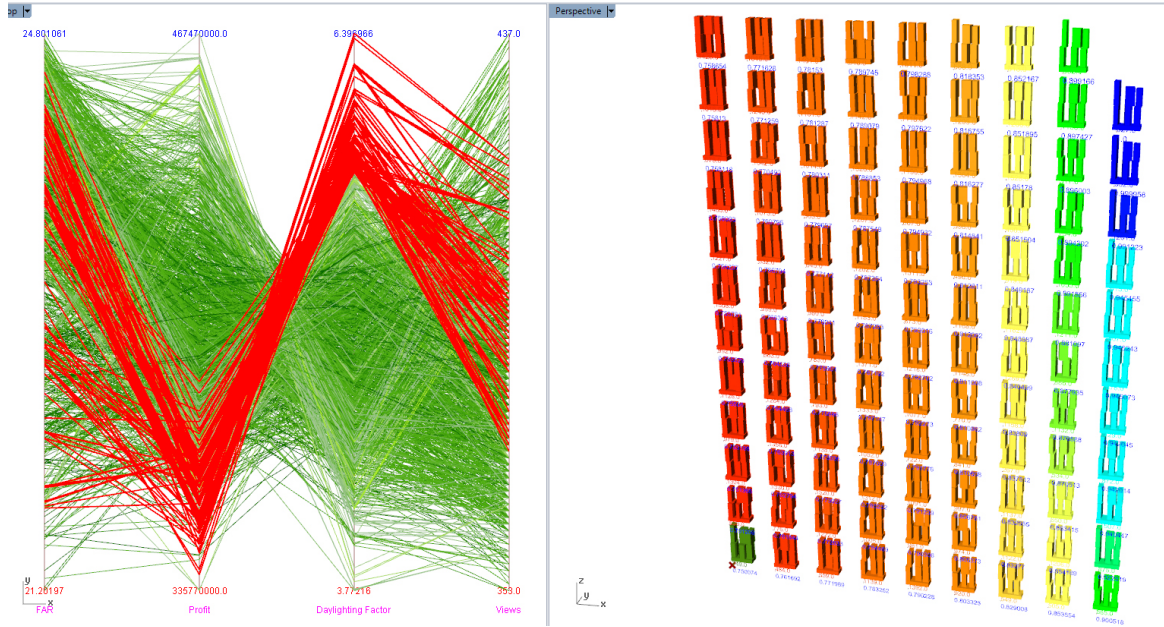


Figure 4-5. Upper quartile of daylight factor of 1<sup>st</sup> optimization run (beveled iteration). (1<sup>st</sup> axis: FAR; 2<sup>nd</sup> axis: Profit; 3<sup>rd</sup> axis: Daylight factor; 4<sup>th</sup> axis: Views).

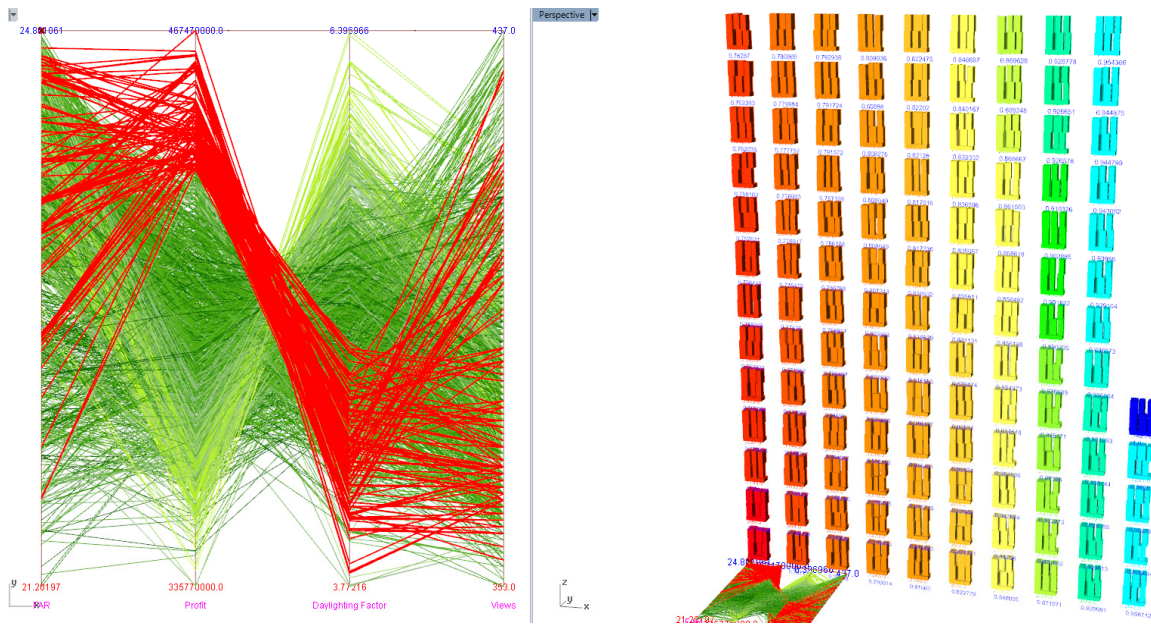


Figure 4-6. Direct relation between the highest FAR and profit that led to low daylight factor (red: worst performer; blue: best performer of daylight factor).

### Second optimization run (stacked blocks)

The second run involved dividing the individual towers into four stacked blocks (Figure 4-3) with the hypothesis of both reducing profit loss through the mitigation of loss of floor area resulting from the beveling and to create a more dynamic looking building.



Most of the solutions including those forming the Pareto front (Figure 4-7) (Appendix C) are spread out mainly between the second and third quartiles of each objective. The daylight factor appears to be an influential limiting objective meaning that its range is not significant in influencing final decisions. Profit, however, will take precedence over other objectives as it has a larger range (341 to 458 million), meaning it will have a larger bias on the choice of solutions over other objectives. Overall, in this optimization run, the solutions follow the same trend as the beveled solutions. Profits, however, did not increase as expected from the change in the formal language, in fact, the minimum and maximum bounds of the profit were nearly the same, even though there was no lose in area (Appendix D).

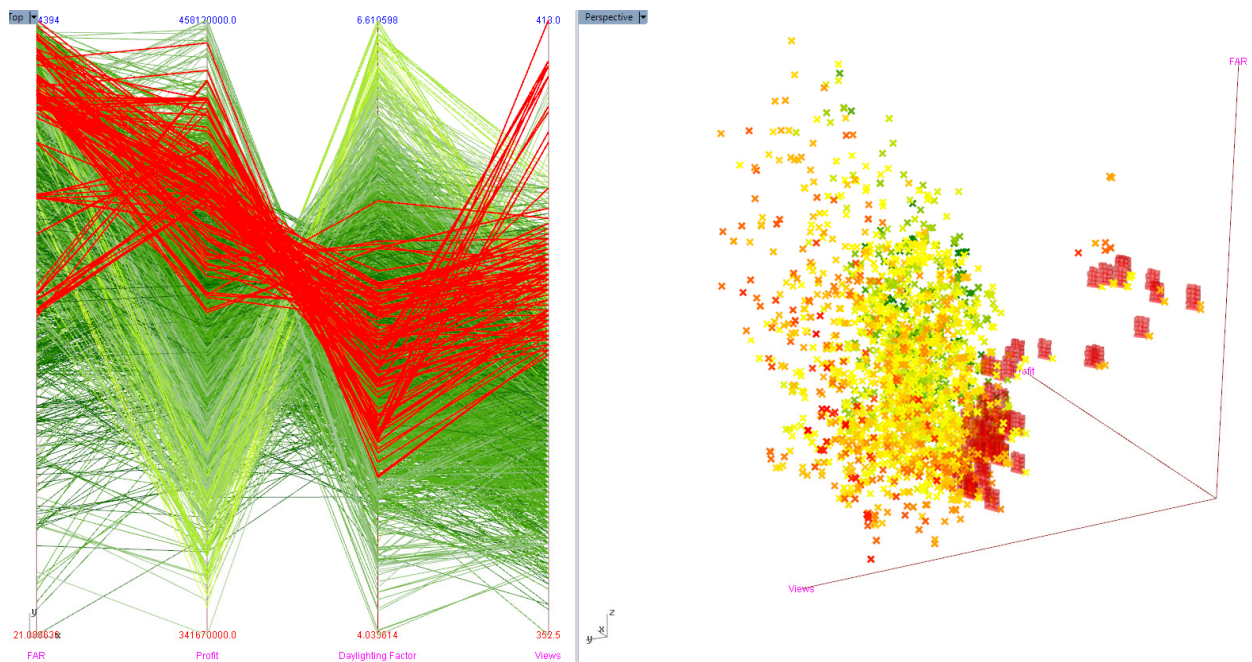






Figure 4-7. Pareto front results of the 2<sup>nd</sup> optimization run (stacked block iteration).

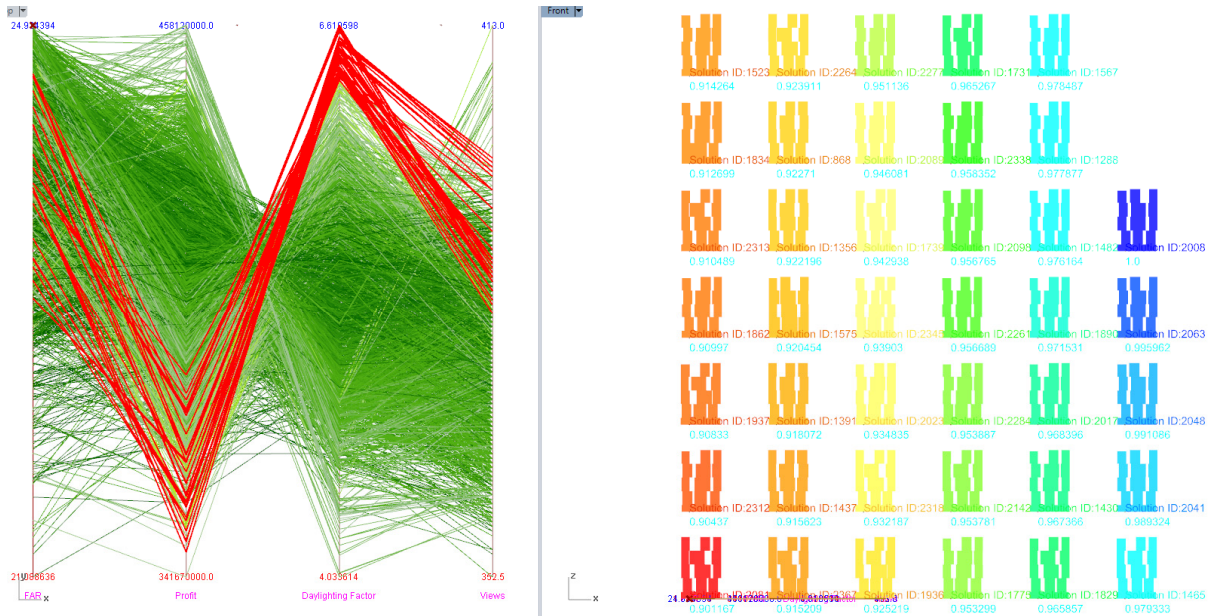


Figure 4-8. 90<sup>th</sup> percentile of daylight factor of the 2<sup>nd</sup> optimization run (stacked block iteration).

The generated 3D forms (Figure 4-8) of the 90<sup>th</sup> percentile of the daylight factor objective are not surprising as the individual blocks are reduced in width and the spacing between them is increased. The profit, however is in the lower quartile due to the significant loss of area from the smaller block dimensions.

#### Comparison between the optimization runs and the De Rotterdam parametric replica

The first optimization run has a Hypervolume Indicator of 0.46, slightly greater than the second optimization run. A possible reason for the slight difference is the number of genes used (Table 4-1). For both runs, the number of generations were increased from the initial assumptions of 20 and 30 generations for the first and second run respectively to ensure convergence of solutions (Table 4-1). The change in the number of generations is representative of the ability to tailor the optimization to the design problem.

*Table 4-1. MOO properties of the De Rotterdam Tower optimization runs.*

	<b>1st run: 4 Objectives</b>	<b>2nd run: 4 Objectives</b>
<b>Hypervolume Indicator</b>	0.46	0.45
<b>Genes (Design Parameters)</b>	60	62
<b>Generations</b>	30	50
<b>Population</b>	1531	2375
<b>Individual run time</b>	30 seconds	57 seconds

The generation of a greater number of solutions leads to improved building performance results as shown in Table (4-2). The results are not conclusive figures because many other domains are not taken into account such as structural and energy use analysis among others. They are, however, representative of the overall trend of the possible design solutions that can be taken into consideration for further design development. The qualitative analysis also led to important correlations and causations between the different objectives such as that some objectives seem redundant (least influential) such as FAR and others may be seen as more influential such as profit.

Table 4-2. Comparison of De Rotterdam optimization runs vs De Rotterdam replica.

	<b>De Rotterdam Replica</b>	<b>1<sup>st</sup> Run (%)</b>	<b>2<sup>nd</sup> Run (%)</b>	<b>Average (%)</b>
FAR	22.43	2.5%	3%	2.75%
Daylight Factor	4.51%	12%	18%	15%
Profit	\$ 355, 443, 885	13%	12.5%	12.75%
Views	372	6.2%	3%	4.6%

The parallels of the trends of the two optimization runs may be because of the same multiple conflicting objectives or that due to hard constraints such as the height restriction led to no significant quantitative differences between the runs. The significant differences in the daylight factor between the two runs does not directly correlate to an improvement in daylight performance for the individual floors as the floor height was not changed. The higher average is because of the resultant forms forming larger perimeters with high values. This emphasizes that the figures are not necessarily representative of the true nature of the final performance.

Views improved significantly in the first optimization run but may result in lower spatial quality due to the beveling of the towers. The stacked block optimization run may not offer significantly improved views but the spatial quality is much better especially considering that offices space makes up 40% of the buildings space. Assumingly, the design intention would be to prioritize office space over other program functions.

### 4.3. Case Study 2: The Bow Tower

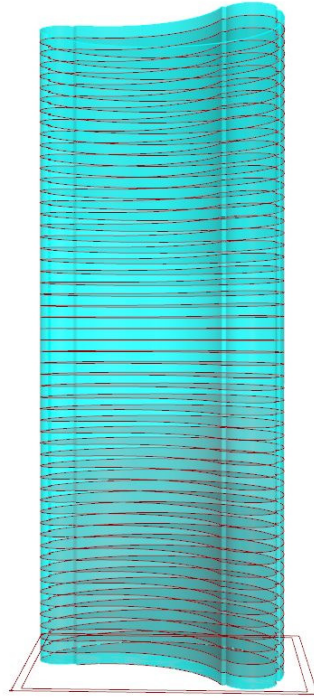
The ‘Bow’ Tower designed by Foster and Partners (Figure 4-9) is used as the second test case to evaluate the effect of the workflow and to critically examine the influence of the number of objectives on the optimization process. The Bow Tower, located in Calgary, Canada, is the tallest structure in Calgary standing at 236m high. The south-facing façade of the tower forms a concave to maximize daylight exposure and heat forming a crescent shaped floor plan. The tower maximizes the perimeter for the north facing offices with views of the Rocky Mountains (Foster,

2013). Other performative design aspects were also important in the Bow's design such as structural performance. However, structural performance is not integrated as part of the parametric definition as the definition was unpredictable which led to incorrect or unfeasible solutions, thus restricting further the possible number of solutions. In addition, the form definition was very simple and the consideration of other performance domains required a more detailed parametric model.



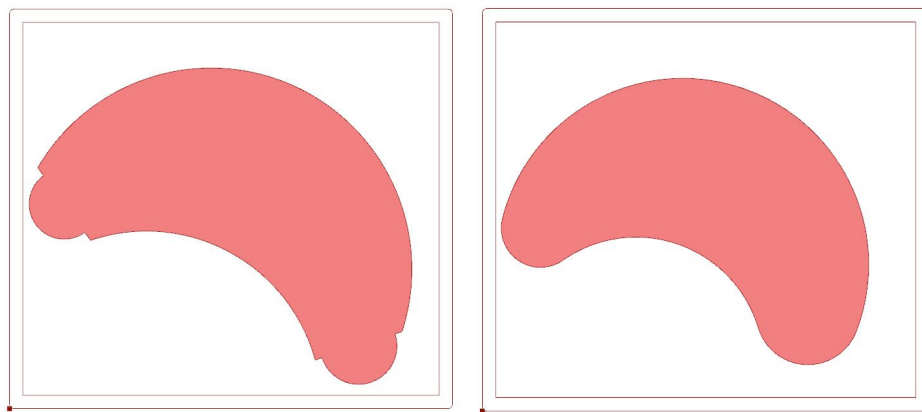
*Figure 4-9. The Bow Tower.*

A replica of the Bow Tower (Figure 4-10) was parametrically modeled using GH (Appendix B) with the exact realistic plot dimensions, building height, number of floors, orientation of the building and the respective areas of the two main program functions: retail and office.



*Figure 4-10. The Bow Tower parametric replica.*

Seven optimization runs were conducted with the aim of increasing the FAR and thus financial profit, average daylight factor, views, and decreasing the shaded area and the glare produced by the building during the summer equinox. These parameters were chosen because they are relevant to this building and its context. Ultimately though like any office tower, the main aim is to maximize spatial efficiency and financial revenue/profit. The same optimization settings such as the mutation rate were used for all optimization runs to ensure consistency.



*Figure 4-11. Left: Same basic geometry parameters as the original Bow Tower design; Right: Side arcs are tangent to the main arcs (modified geometry).*

The optimization runs can be divided into two main categories: three optimization runs using the same parametric setup as the original Bow Tower design consisting of two main arcs joined by two non-tangential arcs (Figure 4-11) and four optimization runs using a slightly different parametric definition of the two main arcs joined by tangential arcs (Figure 4-11). The objectives used for each optimization run are presented in the table below (Table 4-3):

*Table 4-3. Objectives under evaluation for each optimization run.*

	<b>Iteration #1: Original geometry</b>			<b>Iteration #2: Modified geometry</b>			
<b>Optimization Run</b>	<b>1<sup>st</sup> Opt. run</b>	<b>2<sup>nd</sup> Opt. run</b>	<b>3<sup>rd</sup> Opt. run</b>	<b>4<sup>th</sup> Opt. run</b>	<b>5<sup>th</sup> Opt. run</b>	<b>6<sup>th</sup> Opt. run</b>	<b>7<sup>th</sup> Opt. run</b>
<b>No. of Objectives</b>	<b>6 Obj.</b>	<b>4 Obj.</b>	<b>3 Obj.</b>	<b>4 Obj.</b>	<b>3 Obj.</b>	<b>2 Obj.</b>	<b>1 Obj.</b>
<b>Profit</b>	X	X	X	X	X	X	X
<b>Daylight Factor</b>	X	X	X	X	X	X	
<b>Shaded Area</b>	X	X	X	X	X		
<b>Views</b>	X	X		X			
<b>Glare</b>	X						
<b>FAR</b>	X						

The chosen objectives were relevant to the design of the Bow Tower and its context. The bias in the choice of evaluating which combinations of objectives was determined based on the sensitivity analysis carried out from the first optimization run, where conflicting objectives were identified. FAR had a direct relationship with profit and thus the need to evaluate it was unnecessary, hence it was only evaluated in the first optimization run. Glare was viewed as a less important factor in influencing the overall performativity of the design compared to other objectives such as profit, for example.

#### *4.3.1. Model overview: the design loop*

The design generation is carried out by the parametric model (Figure 4-12) where the solutions are evaluated by three Boolean functions to ensure that the generated model does not intersect itself (extended arcs do not intersect), generated model is within site constraints and that a

shadow is not cast on the south side of the Bow River. If the model does not satisfy these Boolean functions, it is discarded and new set of design variables are generated. Once the design satisfies the conditions, a performance evaluation of the solution is carried out. The respective quantitative data and 3D mesh solution are then stored. The amount of generated solutions is determined by Octopus and if the maximum amount of generations (and solutions) have been generated, the optimization process stops.

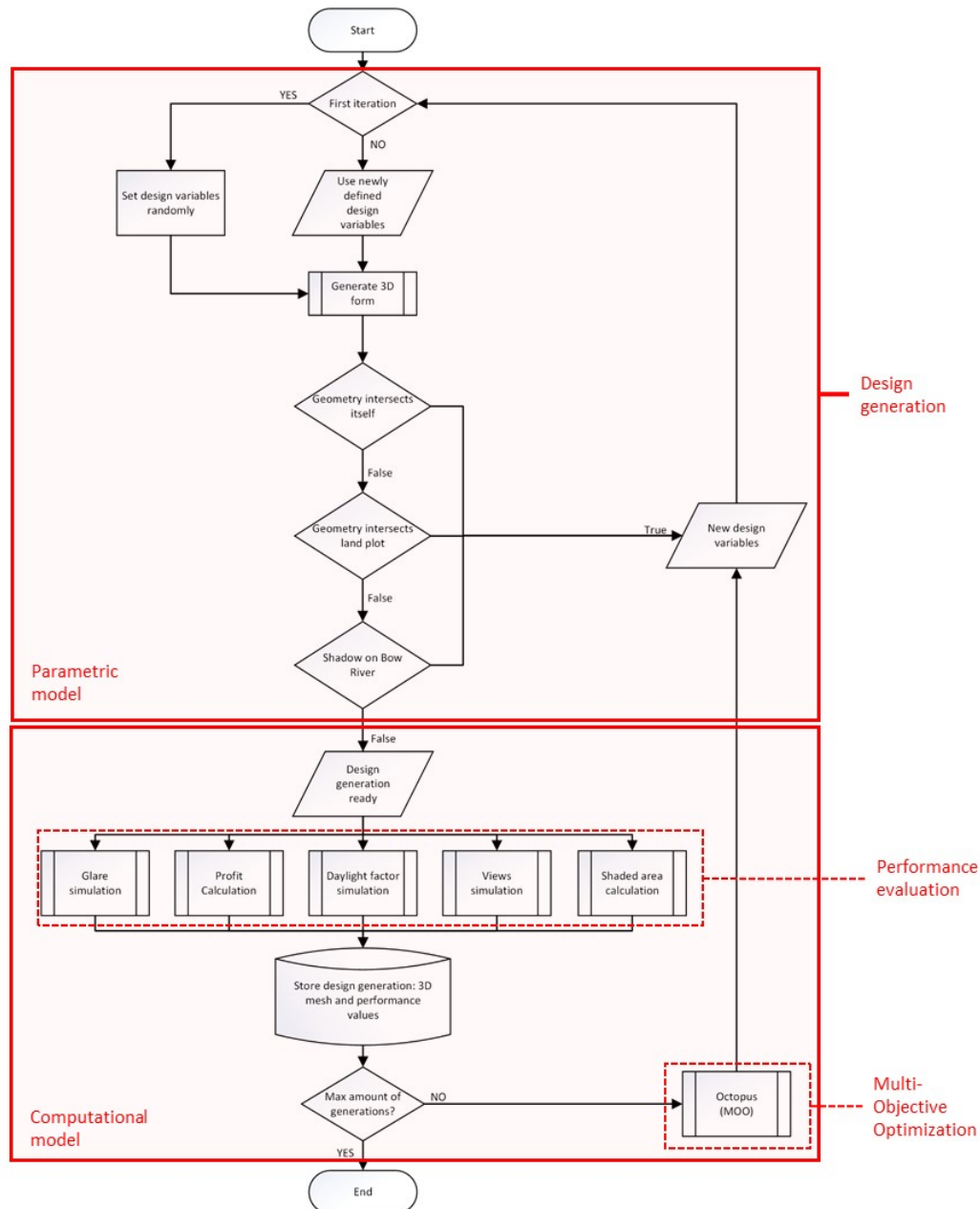


Figure 4-12. Generative model components of the Bow Tower optimization runs.



#### 4.3.2. *The parametric model*

The parametric definition (Appendix B) only has two hard constraints according to the Calgary Land Use Bylaw 1P2007, section 42.1 under ‘Environmental Requirements’, where there is a limitation to where and when a building can cast a shadow on the south bank of the Bow River; and section 42.3 under ‘Discretionary Use Rules’ in which in order to qualify for 20 FAR or more, certain features have to be included in the design as well as consideration of surrounding heritage sites. Environmental requirements are incorporated in GH using a Boolean function to eliminate any solutions that cast a shadow on the south bank of the Bow River during the summer equinox from 10 am to 4 pm. The FAR for the original design is 22 because as part of the development, two heritage buildings: the York Hotel and the Regis Hotel were supposed to be restored and renovated respectively (Calgary Heritage, 2007) as well as other features such as connections with +15 system among other design features.

The financial model is based on the difference between the cost of building the Bow Tower and leasing it on a 25 year lease plan. The profit generated over 25 years is used as the objective and is calculated as per the lease agreement that is between the current owners and the leasing company at a rate of \$36 per square foot at annual rent escalation of 0.75% of office space (HR Reit, 2012) (Appendix B). The financial equation used is as follows:

$$\text{Profit generated} = ((x * ((1 + 0.0075)^{25}) + (x * 25)) - ((\text{Retail area} * \$250) + (\text{Office area} * \$325)));$$

$$\text{Where } x = (\text{gross leasable area} * \$36) \text{ \& } \text{gross leasable area} = \text{building gross floor area} * 0.7$$

The views are calculated using a ‘view analysis’ component from Ladybug that evaluates the visibility of the geometry under test from a set of key viewing points. The surrounding context is modeled accurately in both shape and size to represent the immediate surrounding buildings around the Bow Tower (Figure 4-13). The component evaluates the view from the test points objectively in all directions and outputs the percentage of viewpoints seen by the geometry under test. The key viewing points were the following: the Bow River, Calgary Tower and City Hall.



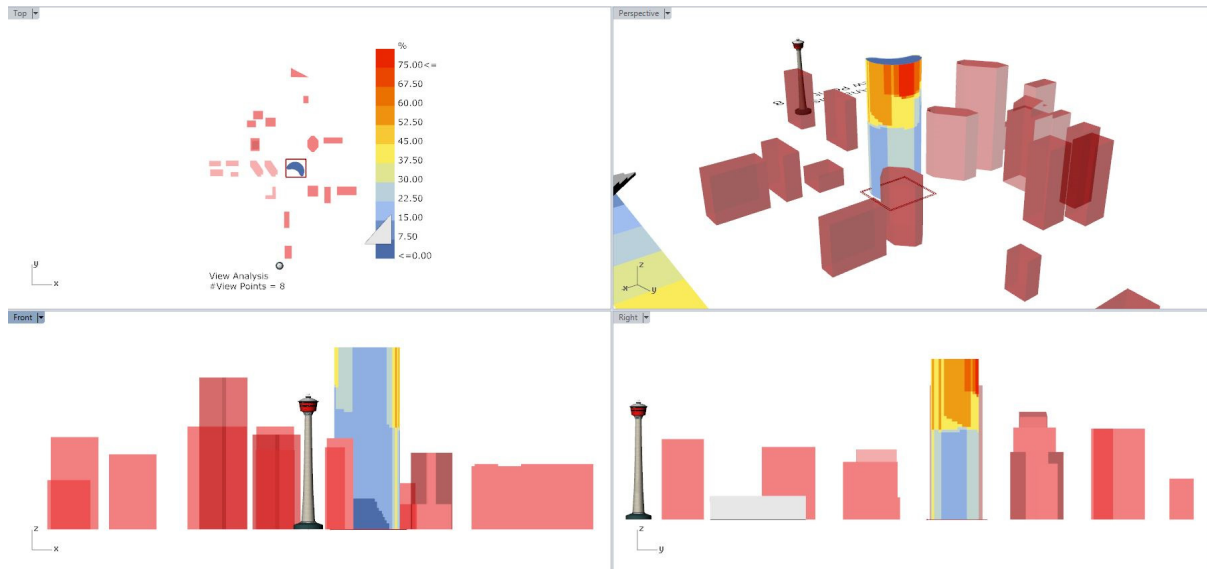
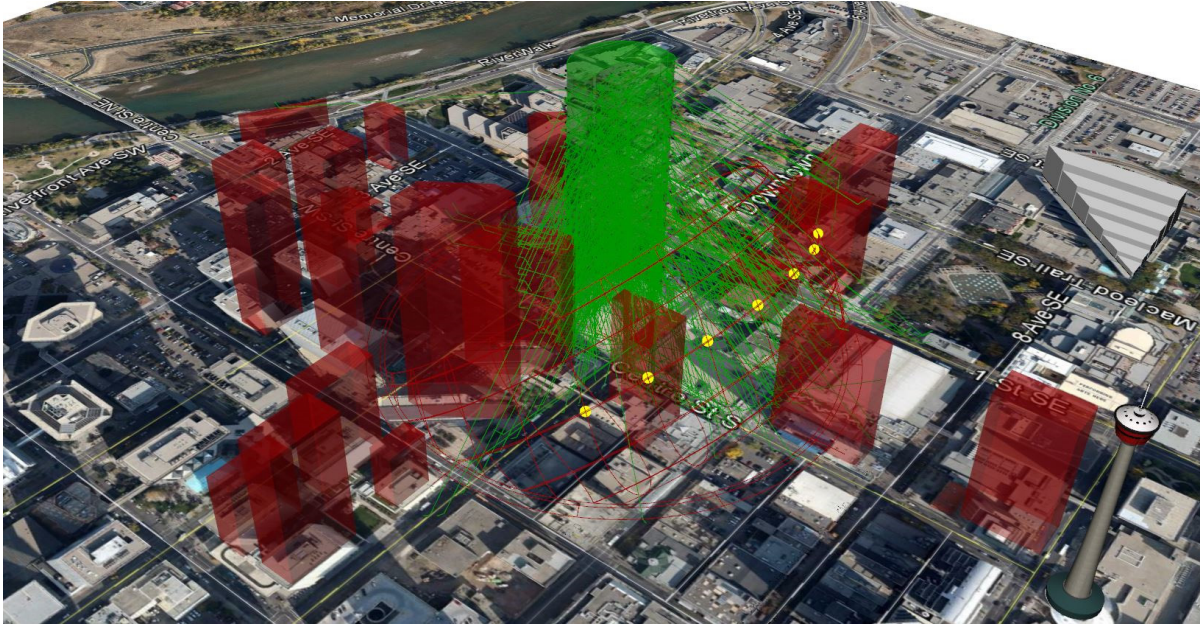


Figure 4-13. Views analysis of the Bow Tower parametric replica (red: best performance areas; blue: worst performance areas).

Daylight factor is defined as the ratio of internal light level to external light level measured in illuminance at the working plane level of 0.762m above ground level. The analysis planes are modeled accordingly at working plane level and modeled on random floors across all the individual towers. The materials chosen for the exterior glazing is ‘double pane – lowE’ with a 65% transmission, generic floor with a 20% reflectance and high reflectance ceiling with a 90% reflectance. All default radiance parameters such as the number of bounces (-ab) are used. The final daylight factor percentage is the average of all these values for the building.

The glare points are tested using ‘bounce from surface’ component from Ladybug which turns the test geometry into a specular surface to reflect sunlight vectors. It is necessary to consider glare as parabolic shapes tend to dangerously focus sunlight at certain times of the day. The number of vectors reflected onto the immediate surrounding public spaces either from direct or indirect sunlight reflection is tallied (Figure 4-14). The evaluation period is also carried out during the summer equinox.



*Figure 4-14. Glare bouncing off the Bow Tower based on the sun vectors during the summer equinox.*

The shaded area is calculated using ‘mesh shadow’ component where a shadow is cast onto a ground plane based on projected sunlight vectors. The total area cast by shadow is summed up for the summer equinox from 10 am to 4 pm according to the bylaws.

#### *4.3.3. Analysis of results*

##### Original geometric definition

The following optimization runs involve the use of the original geometry as the Bow Tower with six objectives: FAR, profit, daylight factor, views, shaded area and glare; four objectives: profit, daylight factor, views and shaded area; three objectives: profit, daylight factor and the shaded area (Table 3). The hypothesis is a larger range of solutions can be achieved with the relaxation of objectives (i.e. less objectives) which ultimately will lead to better performing design solutions. The challenge in improving the performance of an already excellent design proved a difficult task for the MOO engine.

For the first optimization run, the initial assumption of 50 generations proved to be more than needed as the solutions quickly converged due to the multiplicity of the objectives and the search was stopped at 41 generations. The second and third optimization runs, the initial assumption of 25 generations proved to be insufficient and so the number of generations was increased to 30 to

ensure maximum convergence. In total 19 genes or design variables are manipulated in all the optimization runs.

*Table 4-4. MOO properties of the original Bow Tower optimization runs.*

	<b>1<sup>st</sup> Run: 6 Objectives</b>	<b>2<sup>nd</sup> Run: 4 Objectives</b>	<b>3<sup>rd</sup> Run: 3 Objectives</b>
<b>Hypervolume Indicator</b>	0.22	0.52	0.64
<b>Generations</b>	41	30	30
<b>Population</b>	3192	2224	2318
<b>Individual run time</b>	19 seconds	12 seconds	10 seconds

The Hypervolume Indicator (discussed in Chapter 3) was never able to go beyond 0.22 (1 being the maximum possible value) (Table 4-4). The other graphical aids hinted at the inability of Octopus of finding solutions that are more ‘optimized’ (Appendix C). The reduction strategy for the first optimization run was changed from SPEA-2 to HypE as the Hypervolume Indicator appeared stagnant after 10 generations. The reason for changing strategies is to eliminate truncation of the Pareto fronts from generation to generation, thus maximizing the searchable area (increasing the Hypervolume). The Hypervolume Indicator of the second optimization run reached 0.52 and 0.64 for the third optimization which indicates that the solution space was significantly larger for both compared to the first optimization run. The first part of the hypothesis proved correct, as with the relaxation of the search space (i.e. less objectives), a larger range of solutions was found.

Sensitivity analysis of the solutions show that there is a direct relationship between FAR, profit and shaded area. As the form becomes more ‘bulky’, the larger the shadow cast. There is an inverse relationship between FAR and daylight factor and an inverse relationship between views



and glare, and an unclear correlation between views and glare with the rest of the objectives (Figure 4-15). These relationships are reiterated in both the second and third optimization runs.

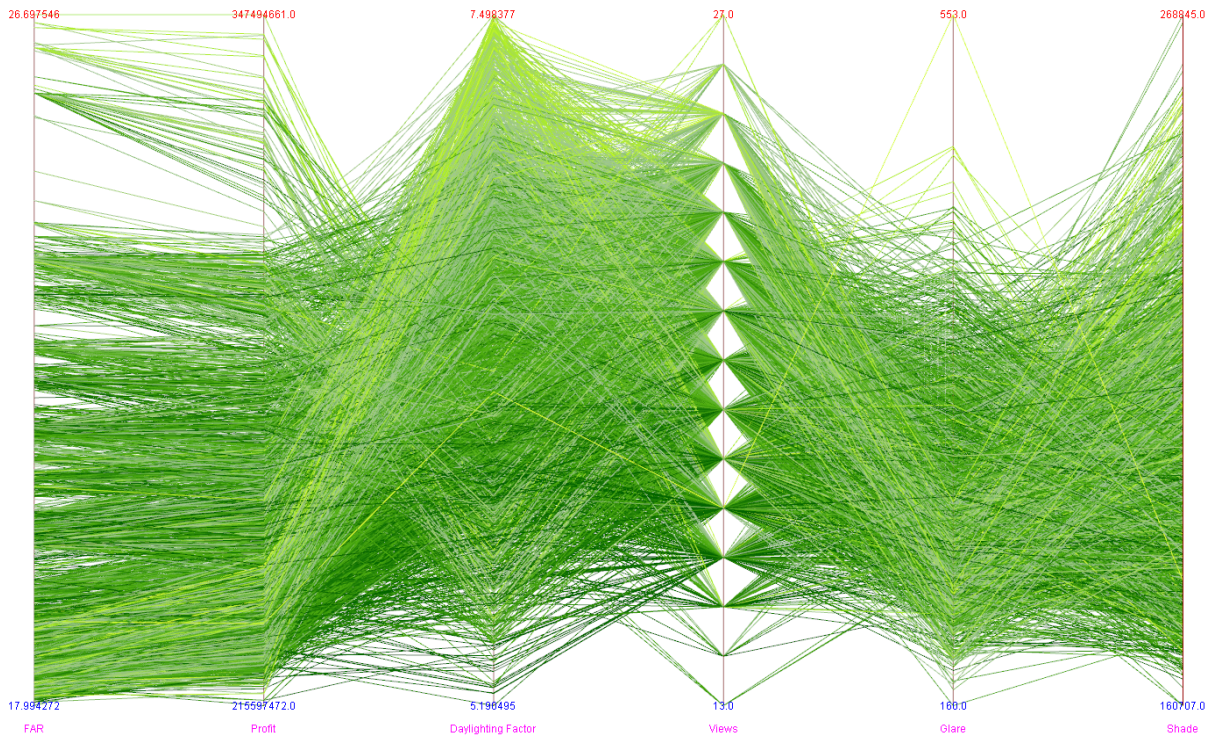


Figure 4-15. Sensitivity analysis of the 1<sup>st</sup> optimization run (6 objectives) (1<sup>st</sup> axis: FAR; 2<sup>nd</sup> axis: Profit; 3<sup>rd</sup> axis: Daylight factor; 4<sup>th</sup> axis: Views; 5<sup>th</sup> axis: Glare; 6<sup>th</sup> axis: Shaded area).

In the first optimization run, the properties of the Pareto front solutions emphasized that indeed no sub-optimal solutions exist (Figure 4-16). Through the use of subset search of the 99<sup>th</sup> percentile of daylight factor, some of the best overall performing solutions are identified. The formal qualities of the solutions are varying with three solutions out nine sharing the same floor plan shape of an ellipse, rotated to both reduce the shaded area and glare while increasing views (Figure 4-17). One of the best performing solutions resulted in an almost elliptical shaped floor plan that seemed to be sculpted by the sun's rays to maximize daylight factor and reduce shaded area whilst maximizing views. Both the 90<sup>th</sup> percentile of profit and the Pareto front results produced rather poor performing solutions both in measurable and non-measurable objectives (Figure 4-18).



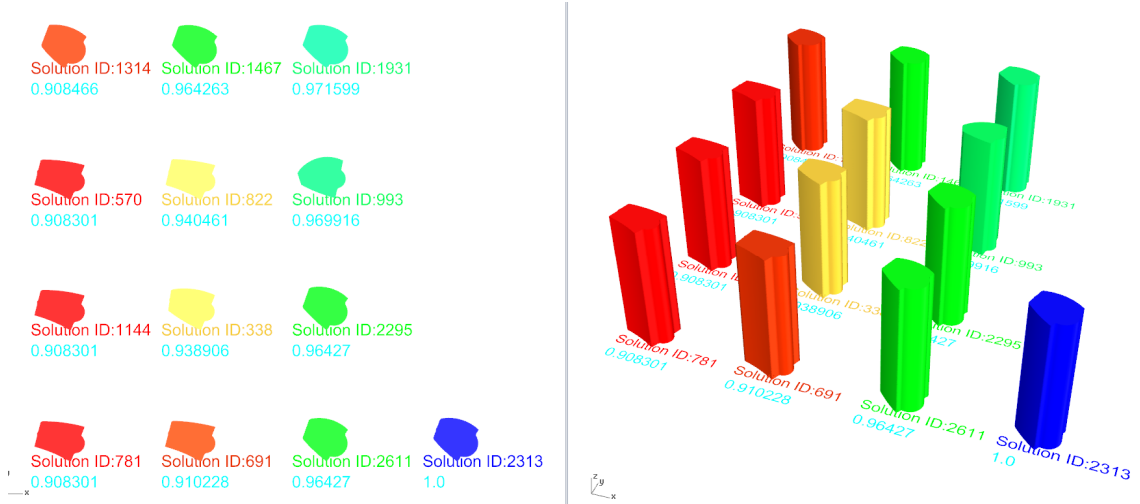


Figure 4-18. 90<sup>th</sup> percentile of profit of 1<sup>st</sup> optimization run (6 objectives).

In the second optimization run, the 90<sup>th</sup> percentile of profit resulted in lower than average daylight factor, average views and above average shaded area (Figure 4-19). The formal language is different compared to the same percentile of the first optimization run. There is a larger variation in the formal definition which arguably results in more interesting forms (Figure 4-20). Pareto front results are spread out mainly in the top half of each domain, which does not necessarily signify that the solutions are ‘well-performing’ solutions especially as the shaded area was high. Upon exploration of the form in the third optimization run, irregular shaped floor plans (Figure 4-21) resulted in higher the normal daylight factor while not being aesthetically pleasing. Overall, the 3D form of the ‘good’ performing solutions share similarities in both shape of the floor plan and orientation to the original Bow Tower design.



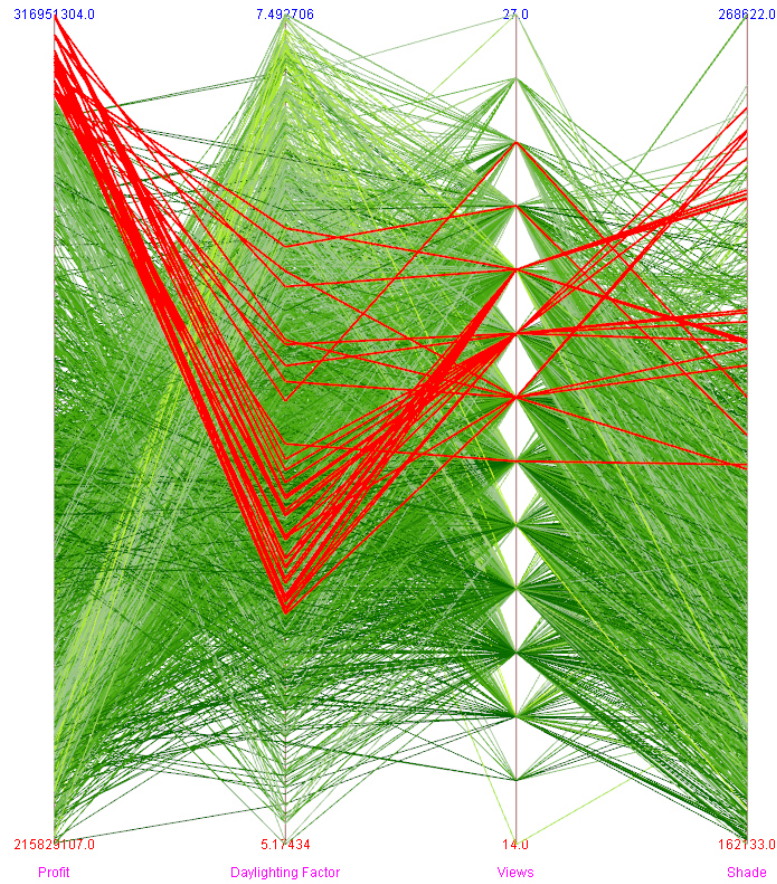


Figure 4-19. Parallel plot of the 2<sup>nd</sup> optimization run (4 objectives) (1<sup>st</sup> axis: Profit; 2<sup>nd</sup> axis: Daylight factor; 3<sup>rd</sup> axis: Views; 4<sup>th</sup> axis: Shaded area).

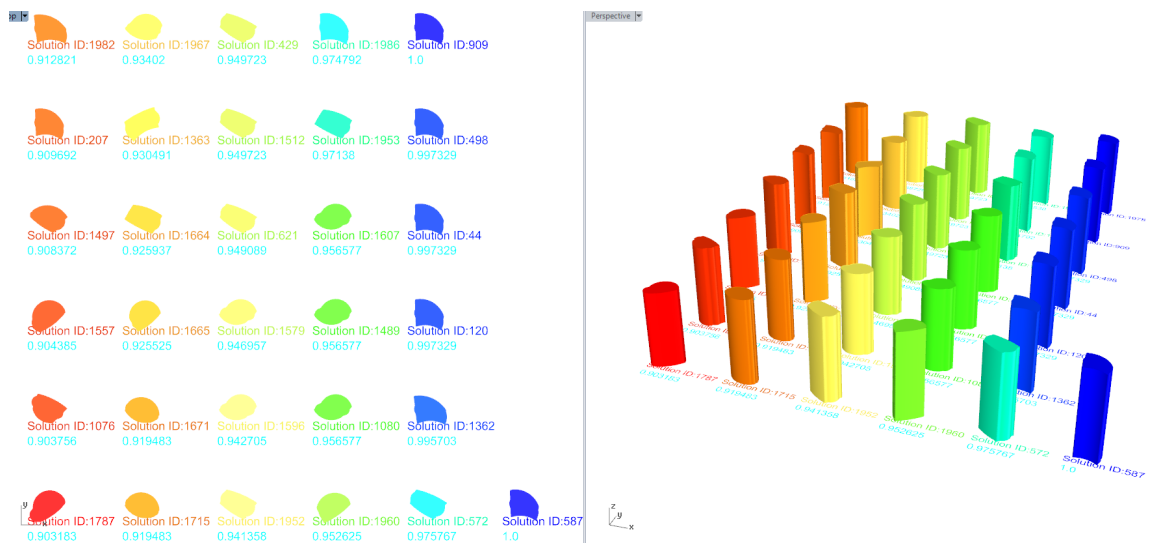


Figure 4-20. 90<sup>th</sup> percentile of profit of the 2<sup>nd</sup> optimization run (4 objectives) – the solutions show a larger range of geometric variety.

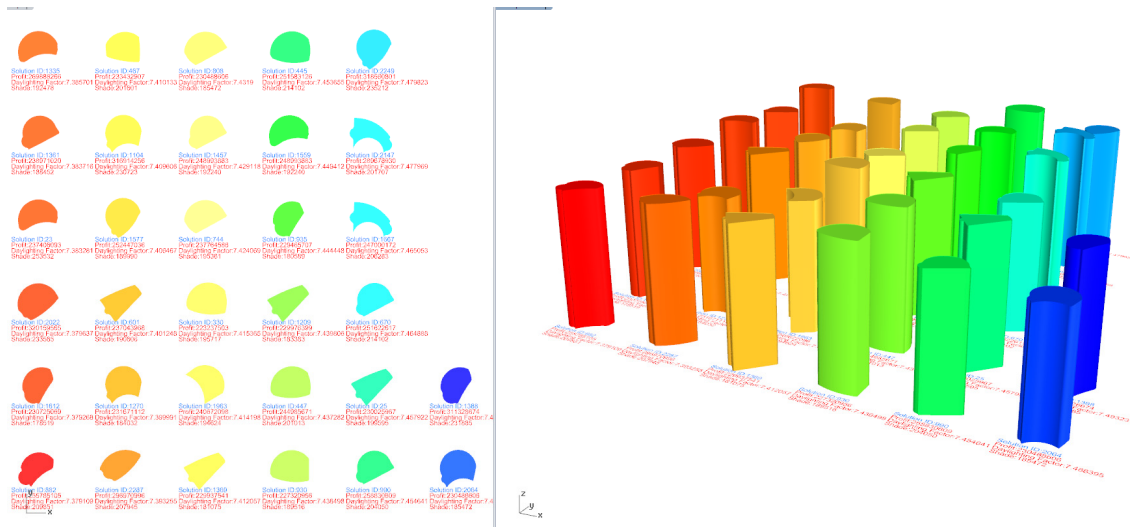


Figure 4-21. 90<sup>th</sup> Daylight Factor of the 3<sup>rd</sup> optimization run (3 objectives) generated irregular shaped floor plans.

### Modified geometric definition

The following optimization runs involved the use of the modified parametric definitions with four objectives: profit, daylight factor, views, shaded area; three objectives: profit, daylight factor and the shaded area; two objectives: profit and daylight factor; and one objective: profit (Table 4-5). The hypothesis is that a different geometric definition would result in better formal qualities of the solutions as an implicit objective. There were many issues facing these optimization runs that are discussed in the next section of this chapter. Convergence of solutions was found quickly by Octopus as there were many Boolean functions that eliminated a significant number of solutions: 5809 solutions in the fourth optimization run, 6864 solutions in the fifth optimization run and 6597 solutions in the sixth optimization run. The Boolean functions eliminated solutions that had geometry intersecting the land plot edges, geometry intersecting itself (extended arcs intersect themselves), and shadow cast by the building on the south side of the Bow River. In total 15 genes or design variables are manipulated in these optimization runs.



Table 4-5. MOO properties of the modified Bow Tower optimization runs.

	<b>4<sup>th</sup> Run: 4 Objectives</b>	<b>5<sup>th</sup> Run: 3 Objectives</b>	<b>6<sup>th</sup> Run: 2 Objectives</b>	<b>7<sup>th</sup> Run: 1 Objective</b>
<b>Hypervolume Indicator</b>	0.46	0.57	0.94	1
<b>Generations</b>	25	25	25	1
<b>Population</b>	1390	1335	1352	100
<b>Individual run time</b>	5 seconds	4 seconds	2.5 seconds	0.5 seconds

The Hypervolume Indicator for the optimization runs are not greater than second and third optimization runs from the previous iteration. The reduced number of genes manipulated may have resulted in a lower Hypervolume Indicator as the possible number of solutions is reduced.

Sensitivity analysis show the same relationships between the objectives exist as in the first iteration. The formal aesthetics is considerably better with this parametric definition and there is a tendency for the solutions to have the same floor plan shape as the original Bow (Figure 4-22 & 4-23). The crescent shape of the floor plan was sculpted by the sun, however, an elliptical shaped floor plan was a recurrent floor plan shape to minimize the shaded area and to maximize the views, yet the profits are significantly lower than those solutions formed like the original Bow Tower. There are certain exceptions that are shaped like the Bow, yet perform better than the Bow (Figure 4-24). The building's concave feature faces south east instead of south west, with improved daylight factor and slightly better views percentage. The views are better score-wise but with this orientation, it does not necessarily mean better views as the building's concave feature faces away from the mountains. This solution also had a smaller shaded area but the profit was slightly less at \$251 million. The overall trajectory of the solutions is very similar to the original Bow.

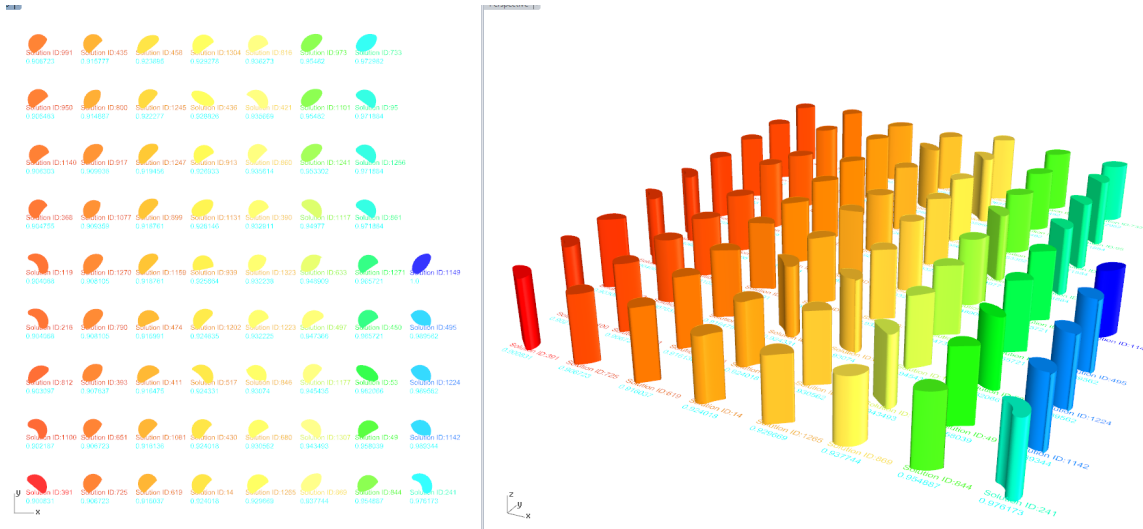


Figure 4-22. 90<sup>th</sup> percentile profit of 4<sup>th</sup> optimization run (4 objectives).

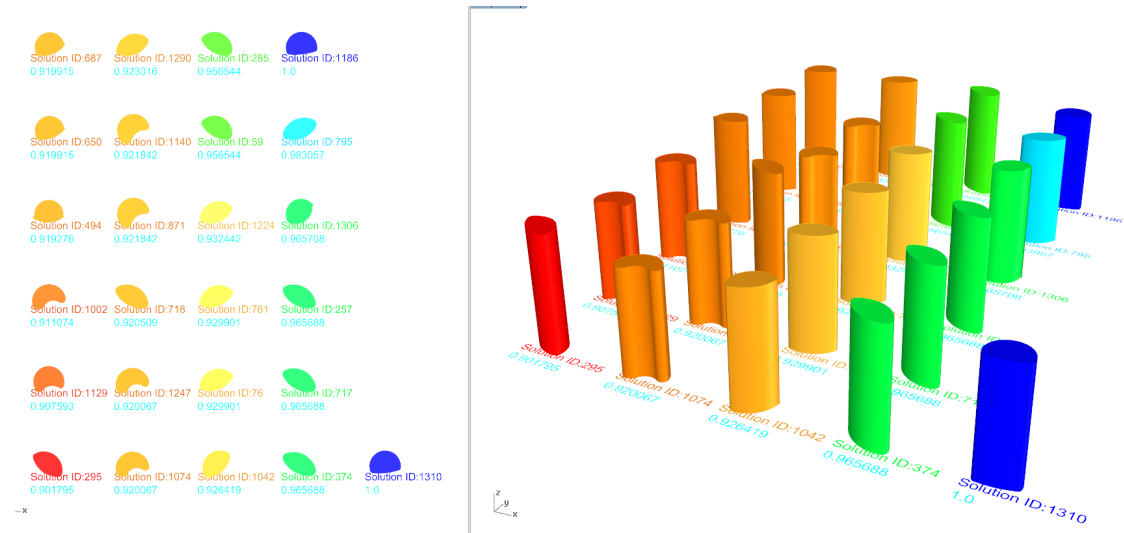


Figure 4-23. 90<sup>th</sup> percentile profit of 5<sup>th</sup> optimization run (3 objectives).

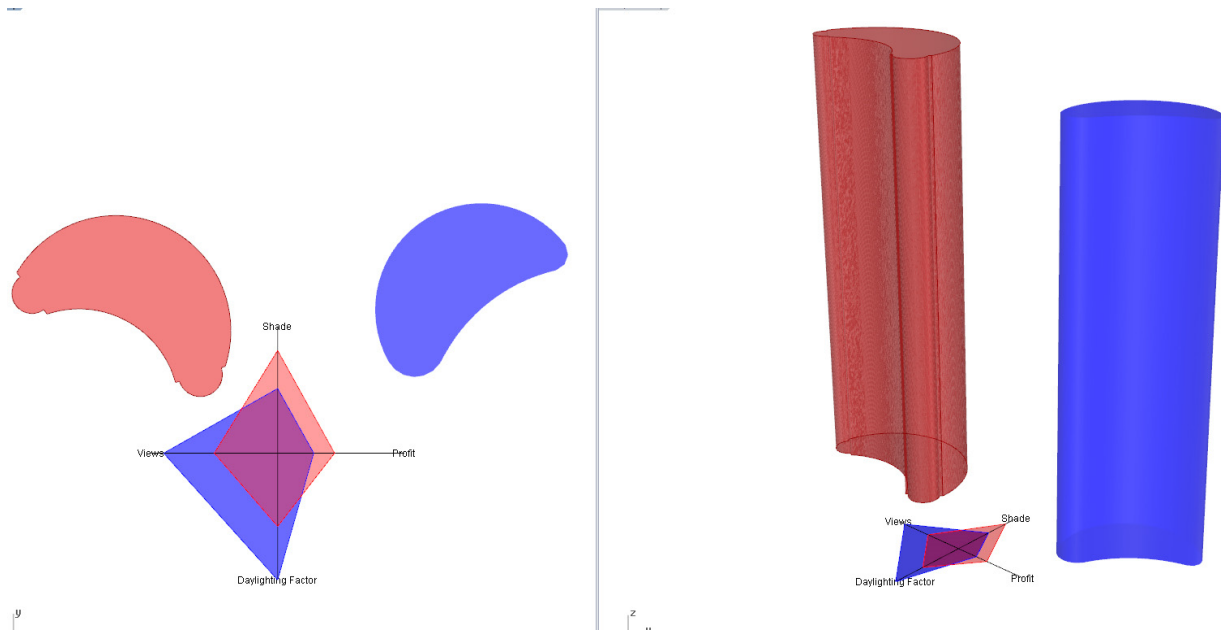


Figure 4-24. Original vs solution ID 254 (overall performs better than the Bow parametric replica).

For the fifth and sixth optimization run, there is a greater number of solutions in the 90<sup>th</sup> percentile of daylight factor than the fourth optimization run (Figure 4-23 & 4-25). The concave feature of the building form tended to face south east rather than south west as of the original Bow Tower design. A comprehensive exploration of the solutions show that this trend is seen throughout different subsets of the different objectives.

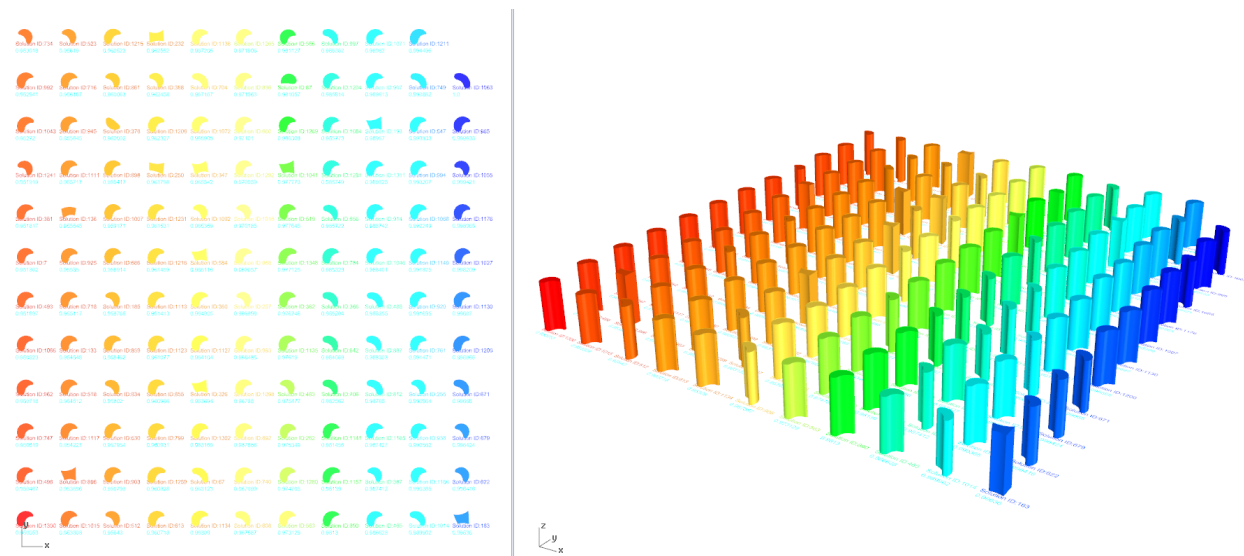


Figure 4-25. 90<sup>th</sup> percentile of daylight factor of 6<sup>th</sup> optimization run (2 objectives): the generated designs are almost the same as the original Bow Tower.

### Comparison between original and modified geometry

The general overview of the results compared to the Bow parametric replica show that there are quantitative improvements (Table 4-6). There are, however, optimization runs that did not have any objectives improve. For the first optimization run, there were improvements across all objectives except for views which remained the same. This may indicate that a greater number of objectives might result in better performing solutions. The second and third optimization runs showed slight improvements in all objectives except profit, which decreased and remained the same for the second and third optimization runs respectively. These results partially prove the hypothesis for the first iteration.

The fourth and fifth optimization runs showed surprising results with the fourth run resulting in improved profit, and reduced shaded area cast, however, daylight factor on average was reduced by nearly 4%. The fifth optimization run only showed improvements in the shaded area but poor performance in daylight factor with a decrease of 6% and lower profit. The sixth optimization run showed considerable improvement in the profit at 4.7% but poor daylight factor performance almost as poor as the fifth optimization run. The seventh optimization run involving only profit as an objective did not show any significant improvements compared to the sixth run with an increase of 0.1% in profit, however it was the best improvement in comparison to the all the other optimization runs.

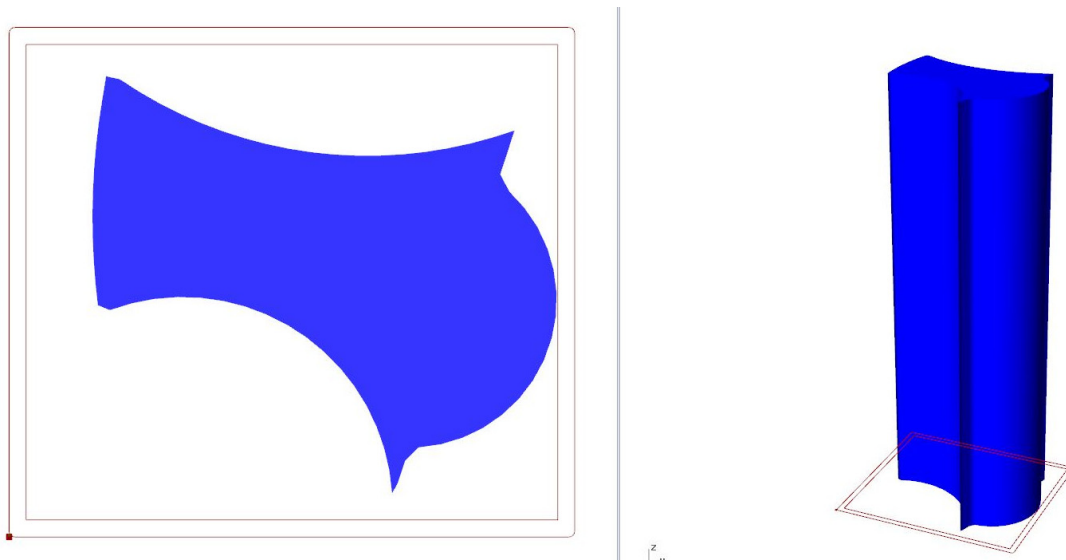
Table 4-6. Comparison of Bow Tower optimization runs vs Bow Tower replica (MG: Modified Geometry).

	The Bow Replica	Iteration #1			Iteration #2			
		1 <sup>st</sup> Opt. run	2 <sup>nd</sup> Opt. run	3 <sup>rd</sup> Opt. run	4 <sup>th</sup> Opt. run	5 <sup>th</sup> Opt. run	6 <sup>th</sup> Opt. run	7 <sup>th</sup> Opt. run
		6 OBJ (%)	4 OBJ (%)	3 OBJ (%)	4 OBJ MG (%)	3 OBJ MG (%)	2 OBJ MG (%)	1 OBJ MG (%)
<b>Profit</b>	\$269,796,148	4.4%	-1.2%	0%	2.1%	-0.1%	4.7%	4.8%
<b>Daylight Factor</b>	6.2%	2.4%	2.2%	2.2%	-3.8%	-6%	-5.7%	
<b>Shaded Area</b>	240770 m <sup>2</sup>	-12%	-11%	-15%	-13%	-12%		
<b>Views</b>	20%	0%	2.5%		0%			
<b>Glare</b>	429	-20%						
<b>FAR</b>	21.44	4.4%						

There are no parallels in the results between the first iteration and the second iteration, even though there were no significant changes to the base geometry. The results indicate a surprising trend: the more objectives, the better the *overall* performance of the solutions. It is counter intuitive that more objectives would lead to better performing solutions, as indicated by the Hypervolume Indicators that the less objectives to optimize for, the larger in value it is.

However, this is not a conclusive statement as there are multiple factors that are not considered and the forms generated may not perform better from a formal aesthetic point of view. A more comprehensive exploration of the 3D solutions reveals that even though some solutions did exhibit improved quantitative results, there are strange ‘mutations’ of the shape of the floor plans. Through a subjective derivation these solutions indicate that they cannot be used for further design development as their spatial properties are poor or they would not work well in the

given context (Figure 4-26). These solutions may have improved the overall average performance but the values are only indicative until further extensive exploration is carried out.



*Figure 4-26. Strange parametrically generated solutions that cannot be used for further design development even if they perform well.*

#### **4.4. Analysis and Discussion**

The developed workflow with its associated tools played a crucial role in identifying multiple issues. First and foremost, the Pareto graph in Octopus does not always show potential errors or deceptive bottlenecks in the optimization runs as a result of the parametric definitions. The sensitivity analysis carried out especially using the parallel-plot graph, highlight strange or unpredicted relationships and thus making the trade-offs of the solutions on the Pareto graph more understandable.

The trial runs of the optimization of the two case studies and their variances show that poor parametric definitions can greatly influence the final results. A poor understanding of the GH components and how they work can lead to strange results which, without sufficient experience, can lead to poor design decisions. The lack of knowing what the parametric definition could generate requires at first quick test runs to comprehensively understand how the definition works and, the limitations of each of the parameters and how they correlate. With this in mind, the flexibility of Grasshopper in modifying parametric definitions and the ability of Octopus to tailor the optimization strategies based on the design problem may have arguably yielded better performing solutions (Vierlinger & Hofmann, 2013).

The lack of experience using such software may lead to several scenarios such as the dichotomy or stratification of results within one objective or large ranges which occur due to bad geometry or poor mesh divisions. It is therefore essential to predict how a design will perform and within what ranges so not to be tricked by the software with ‘impressive’ results.

The Hypervolume Indicator of all the optimization runs show the same trend: the greater the relaxation of the number of objectives and the greater the Hypervolume Indicator, the easier it is for Octopus to find a wider range of solutions (greater extremes of each objective). Hard constraints such as building height, shadows cast limitations or others, limited the range of possibilities of the solutions which may have performed better in the various domains. A significant population was eliminated in the second case study due to various hard constraints by the geometry and the bylaws.

The De Rotterdam Tower optimization runs included a greater number of parameters (60 and 62) compared to the Bow Tower (19 and 15) which also influences the range of the possibility of solutions but not the Hypervolume Indicator. In other words, a new Hypervolume is defined for that specific design problem by finding its extreme values; therefore the Hypervolume Indicator values cannot be compared based on changes in the number of design parameters.

The change in the parametric definitions for both case studies expanded the horizon of discovering interesting sub-optimal designs. The ability of the developed workflow and tools in helping the designer better understand the limitations of the site and the design proved fruitful. The encouragement of exploring multiple alternative parametric definitions may not necessarily lead to better quantitative results as shown in the Bow Tower case study but may lead to more interesting form qualities.

There is also an interesting trend in the solutions: similar performing solutions shared the same number of vertices and faces in their solution meshes even though they have slightly different formal results. This may give greater insight into how Octopus works but it could also be a coincidence.

#### **4.5. Analysis Conclusion**

The use of the information visualization tools helped make sense of the prescriptive information produced with the 3D solutions. The tools help identify objectives that at first seem necessary

and are later neglected in order to refine the search and optimization run times. Parameter ranges and choice of parameters can be redefined by investigating the solutions produced via Octopus by reinstating the solution. Qualitative analysis of the produced solutions are essential in identifying potential pitfalls or limitations of the choice of geometry as well as further understanding the trade-offs of the solutions. Formal qualities, as an objective, was included in an implicit fashion following the optimization runs and helped determine whether the quantitative performance metrics were accurately representative of the notion of a ‘good’ solution or not. In order to contextualize the solutions that are to be explored, it is necessary to first exploit and identify the ‘best’ possible solution(s) for a given design. Exploitation and exploration therefore have a symbiotic relationship and even though one of the aims of this thesis was to encourage exploration of the solution space, exploitation is equally important.

The use of already built designs helped contextualize MOO based results and identify whether solutions are indeed improving performatively or not. The results show that there is no correct answer, as in general, some iterations did show improvements and some did not. The case studies also show that there are no limitations to the type of geometry that can be explored due to the flexibility of GH.

The ‘creative optimization workflow/tools’ enable qualitative interactions by the inclusion of formal aesthetics as an implicit objective. The evaluation of the formal qualities in the optimization process led to a different formal exploration runs in both case studies. It is the difficulty in quantifying aesthetics or other qualitative objectives, that a decision maker is necessary in identifying potentially bad solutions or optimization trajectory. The tools aid the designer to contemplate multiple solutions and substantiate any optimization decisions such as changes in the mutation strategy as in the Bow Tower (6-objective) optimization run which may lead to alternative solutions. The overall process is analogous to the design process itself and in this case, where MOO is used, the design decisions are not solely based on quantifiable metrics such as financial performance.

One of the main goals of this research is to reduce the design cycle latency through the integration of parametric design, MOO, simulation, and feedback loops. By using and tightly integrating off-the-shelf software and by semi-automating the workflow using a common platform, some of the commonly encountered data exchange and interfacing issues were avoided.



The design solution exploration, however, increased the design latency which would affect the overall design cycle latency. In other words, design cycle latency was reduced from one perspective but increased from another.

## Chapter 5: Conclusion and future work

### 5.1. Conclusion

The lack of development of Multi-Objective Optimization (MOO) tools for the purposes of design and the lack of development of interactive information visualization tools have limited the use of MOO in design. Yet, with the significant improvements in both algorithms and computing power, this should not remain the case. The potential of this workflow presented in this thesis shows that even though MOO is unpredictable in the broad sense, the exploration of several thousand solutions is feasible. The first two parts of the thesis hypothesis stated that the combination of MCDO and information visualization tools will help improve the understanding of the quality of the design solution space, and to enable the designer to evaluate the formal qualities of a design. These have been proven and show the potential of the workflow's success through running several optimization runs.

The impact of the developed workflow was examined through two case studies. The retrospective design of two already built buildings was conducted to outline the benefits of the combinatorial power of MOO and information visualization tools. The resulting designs show improvements in both the quantitative and qualitative aspects compared to the original designs. The third part of the thesis hypothesized that the solutions would be multiple performative, yet not all performance domains were considered as part of the evaluation process such as structural or energy performance. Since many other important performance domains were not taken into account, the hypothesis was at best partially proven, i.e. its validity was only partially demonstrated. It is very difficult to predict if the performance of the solutions, from both a quantitative and qualitative perspectives, would have improved (or not) if other domains had been taken into consideration.

One of the objectives of this thesis is to make MOO based solutions more accessible during the early stages of design by aiding the decision making process using a comprehensive set of quantitative and qualitative objectives, and to encourage optimization in a creative fashion. The use of already well-defined objectives and geometric parameters is a limitation as explained in the next section of this chapter. As pointed out by other researchers, ill-defined objectives take several iterations to define well (Turin *et al.*, 2011). The approach in this thesis of redefining both the objectives and geometric parameters based on the runs is analogous to the conceptual

design process of constantly reiterating to better the overall design. In other words, learning with the system leads to a refined understanding of both objectives and geometric parameters, and the differences that result in different performance metrics.

The workflow involved the use of multiple plugins for various simulations such as DIVA for daylight factor simulation. The tools are all based on open source software which have been verified (Reinhart & Breton, 2009). The concept of the workflow is not novel, as presented in Chapter 2 through the various precedents. The effectiveness of the workflow can only be directly compared to *ParaGen*, which was developed in-house at the University of Michigan (Von Buelow, 2011). The software is not available to the general public which hinders the possibility of a real-time operative comparison. The disadvantage of *ParaGen*'s workflow is in the inability to explore solutions in 3D form unless through a tedious process of exporting a VRML file for each individual solution.

The workflow presented in this thesis is novel in Grasshopper3D, and has been made feasible by Octopus, the MOO engine. The workflow is extensible as it is based in Grasshopper3D and thus open to additional parameters and performance objectives. Octopus is a relatively new plugin developed late in 2012 and has been since further updated with features over the past two years. Octopus has been broken into several components which can be further customized to the design problem via a new plugin called Octopus.Explicit (Octopus.E) (Vierlinger & Bollinger, 2014).

The performance of the solutions of the two case studies did show improvements compared to the original design. The De Rotterdam case study showed very significant improvements in the performance of the objectives, with daylight improving substantially at an average of 15%. The Bow Tower case study showed improvements across all measurable objectives in the first optimization run that considered six performance objectives. In the second and third runs, the improvements were mediocre, if any. Slight changes in the geometry and in the number of parameters in the second iteration resulted in lower average daylight factor. This phenomenon may be attributed to problems in the parametric definition. The fifth optimization run was the worst overall performer compared to the rest of the optimization runs of the Bow Tower. The sixth optimization run showed considerable improvement for profit but a poor performance for daylight factor. The seventh optimization run for a single objective showed a minuscule

improvement of 0.1% for financial performance compared to the sixth optimization run, however, it was the best financial performance result compared to the other optimization runs.

The relaxation of the number of objectives did result in higher Hypervolume Indicator values as presented by the first iteration of the Bow Tower. A higher Hypervolume Indicator implies that Octopus can find a wider range of solutions (greater extremes of each objective). Hard constraints limited the range of possible solutions thus eliminating potentially ‘good’ solutions that may have been worthwhile pursuing.

#### *5.1.1. Scope and limitations*

The science of MOO and the use of Octopus include many different parameters and are not just limited to the choice of reduction and mutation strategies as outlined in Chapter 3. Octopus offers the ability to manipulate various parameters such as the elitism ratio, mutation probability and rate, and crossover rate. These parameters can diverge or converge the possible range of solutions that can be generated. These parameters were kept identical for all the optimization runs to ensure consistency in the optimization process, thus the focus of this thesis was to explore the impact of the developed workflow and tools. According to Verilinger & Hofmann (2013), these parameters can have an influence on the optimization results which ideally may lead to unexpected or interesting design solutions.

Three limitations are discussed below: technical difficulties with the software, the use of already built designs and hence parameters, and the use of a limited number of performance domains.

The technical difficulties of using Grasshopper3D is in its inability to handle large amounts of data. Excessive use of memory during the optimization runs led to slow runs and sometimes crashes of the software. Part of the setup was also trying to streamline and optimize the parametric definition in such a way that data trees do not grow substantially. It was also necessary in limiting the ‘detail’ of the geometry to basic elements to reduce simulation run time. Grasshopper3D as a tool is not designed to deal with or store large amounts of data. Grasshopper3D will move its platform to support 64-bit computing and parallel processing, hence augmenting the performance of MOO runs and the ability to handle large amounts of data. Essentially, with a more powerful GH, overall design latency will be reduced even more notably in the near future.

The retrospective design of both the De Rotterdam Tower and the Bow Tower narrowed the scope of exploring ill-defined objectives and geometric parameters. To fully understand the implications of the workflow on the early stages of design, a hypothetical design problem had to be articulated. The different parametric definitions for each case study gave a vague understanding of the potential of the workflow as well as the pitfalls if applied during the early stages of design. The choice of the parameters limits the exploration of widely different forms and thus narrows the scope of really exploring different alternative performative designs. It is also important to directly state that using parametrics or MOO in design is not a replacement for a designer, but rather a very powerful aid for exploring and understanding the design problem.

A multiple performative design is defined in Chapter 2 as a design that performs well on several fronts whether quantifiable or not. A limited set of performance domains were considered for this study due to both computational and time limitations. In addition, the parametric models were simplified which thus limited the feasibility of including other objectives. Not all domains are equally important but some are essential in developing a clear understanding of the limitations of a design. Ideally, in a computational setup that includes all possible simulation domains is very rare due to the above limitations. It is therefore important from the outset to identify which objectives are potentially influential on the design.

## **5.2. Future Work**

The proposed workflow is designed to enable optimization in a creative, flexible, interactive manner. The workflow combines MOO and the developed information visualization tools in Grasshopper3D to make MOO designs more accessible. Several factors were not considered as part of this thesis such as the other parameters in Octopus, unknown or undefined geometric definitions, other performance domains such as structural analysis and the various scales of application in terms of both size and details.

Change in the parameters in Octopus may have yielded better performing solutions, faster optimization runs and more extreme solutions. The range of possibilities would change the way in which a designer contextualizes the generated solutions. The influence of such parameters would be documented to set forth a guideline for the use of MOO in design. There is a significant gap in the literature about the use of MOO for design problems as MOO engines have not been readily available or accessible for designers. It is only recently in the past decade that

the use of optimization in design has become more accepted by the designers with tools such as Galapagos for Grasshopper3D and improved software linkages with simulation tools.

The use of the proposed workflow and its associated tools in the conceptual stage of the design process needs to be evaluated using a hypothetical case similar to the one used by Lin & Gerber (2013). The conceptual stage of a design is characterized for its unknowns and wide spectrum of uncertainties. The proposed workflow may prove to be successful at this stage of the design process as shown through the implementation of various parametric definitions in this thesis. An interesting application would be to test the workflow and tools in a design studio environment to verify the potential of the workflow as well as assess whether design latency is reduced compared to the conventional design workflow (Gerber & Flagger, 2011; Lin & Gerber, 2013).

Other domains such as energy or structural analysis can be included as part of the optimization process due to the flexibility of Grasshopper3D. These inclusion of these domains may lead to better performing solutions and will help estimate the trade-offs on each solution from early on in the design process. The selection of the domains would be determined by the design problem and aims.

The flexibility to parametrically model any kind of geometry using Grasshopper3D offers an advantage in tackling differently scaled problems. Optimization is known for its power when used as an exploitation tool. The power of combining both ends of the spectrum, exploration and exploitation, offers a more comprehensive creative optimization process. For example, a MCDO can be carried out for a building façade considering its materials, cost and energy performance which will be influenced by the MCDO results of the general 3D form of the building. The general 3D form is also determined through the optimization of performance domains such as structure, daylight factor and energy analysis. A dialogue can be established between both the ‘detailed’ optimization and the ‘general’ optimization which mutually benefit and inform from each other. The conceptual design stage will influence the detailed design stage and vice versa from the beginning establishing a new form of interrelationship between scales of process and form. Design latency could be reduced significantly across all stages of design while ensuring improved performance across multiple domains and design stages.

Moreover the full automation of the overall workflow and the creation of more intuitive ‘smarter’ interactive interface will help present information when it is needed. This is necessary

in mitigating information fatigue which is equally as redundant as having no information at all and increases the chances of making poor decisions.

### **5.3. Summary**

In conclusion, the use of MOO in design has been criticized as a method that polarizes design into science vs art, quantitative vs qualitative, numbers vs intuition. This thesis has approached both poles as necessary elements for a successful performative design. Through the use of MOO and information visualization tools, both the quantitative and formal aesthetics were considered. This dynamic relationship between both elements directly resulted in the formation of biases of objectives during the exploration of the solutions, as one objective (including formal aesthetics) can direct the search into unexpected trajectories.

The purpose of the ‘creative optimization workflow/tools’ is to provide a flexible real-time feedback loop along with 3D geometric representation, in order to influence early design decision making from both a quantitative and qualitative perspective as well as the aesthetics point of view. The tools offer improved accessibility to MOO results for the designer and the ability to search, sort, filter, comprehend the solution space with the consideration of overall design performance. Grasshopper3D provided a common platform for added functionality in design development by the combining of multiple disparate domains via free plugins such as DIVA according to the designer’s preferences and cognitive abilities.

The potential of the described workflow is significant in the early stages of the design. It can be made more efficient, smart and flexible with the use of additional tools/components in Grasshopper3D. The intent of these tools/components is to allow the designer’s to generate complex geometry, explore and evaluate a greater number of alternatives quickly through the integration of multiple domains and to reduce overall design latency. Too many solutions may devalue the understanding of the design solution space; the tools will thus aid the designer in the overwhelming task of comprehending and cognitively determining which solutions are suitable.

Due to the significant criticism levied on the use of MOO in design, research about its impacts have been limited to specific architectural problems (for example, structural design of trusses). This thesis unveils the feasibility of conducting and expanding the potential of using computational power to aid the design process, to make better informed design decisions,

encourage a more collaborative integrated approach to design and to improve overall performance of buildings.



## Bibliography

- 44 Floors. (2015). *Apartments*. Retrieved from 44 Floors:  
[http://www.44floors.com/index.php?mod=app\\_finder&section=Apartment%20finder&id=177](http://www.44floors.com/index.php?mod=app_finder&section=Apartment%20finder&id=177)
- Abraham, A., Jain, L., & Goldberg, R. (2005). *Evolutionary Multiobjective Optimization: Theoretical Advances and Application*. London: Springer.
- Aish, R., & Woodbury, R. (2005). Multi-level interaction in parametric design. *Smart Graphics* (pp. 151-162). Berlin: Berlin Heidelberg: Springer. doi:10.1007/11536482\_13
- Andersson, J. (2001). *Multiobjective Optimization in Engineering Design*. Linköping: UniTryck Linköping.
- Andrew, V. (2005). Form Follows Data: The Symbiosis between Design & Information Visualization. *CAAD Futures*, (pp. 31-40). Vienna, Austria.
- ArchDaily. (2013, November). *De Rotterdam/OMA*. Retrieved from ArchDaily:  
<http://www.archdaily.com/?p=451377>
- Bader, J. (2009). *Hypervolume-Based Search for Multiobjective Optimization: Theory & Methods*. Zurich: ETH Zurich. Retrieved from  
<http://johannesbader.ch/files/ThesisWithCover.pdf>
- Beesley, P., & Bonnemaïson, S. (2008). Why Revisit D'Arcy Wentworth Thompson's On Growth and Form? In S. Bonnemaïson, & P. Beesley, *On Growth and Form: Organic Architecture and Beyond* (pp. 7-13). Canada: Tuns Press and Riverside Architectural Press.
- Benjamin, D. (2012). Beyond Efficiency. In S. Marble, *Design workflow* (pp. 14-27). New York: BIRKHÄUSER.
- Bittermann, M. (2009). *Intelligent Design Objects (IDO) - A Cognitive Approach for Performance-Based Design*. Delft: TU Delft.
- Bradner, E., Iorio, F., & Davis, M. (2014). Parameters tell the design story: ideation and abstraction in design optimization. *SimAUD 2014*, (pp. 77-84). Tampa, Florida.

- Braham, W. (2005). Biotechniques: Remarks on the Intensity of Conditioning. In B. Kolarevic, & A. Malkawi, *Performative Architecture: Beyond Instrumentality* (pp. 57-70). New York: Spon Press.
- Burry, M., & Murray, Z. (1997). Architectural Design Based on Parametric Variation and Associative Geometry. *15th eCAADe Conference: Challenges of the Future*. Vienna (Austria).
- Caldas, L. (2007). Generation of energy-efficient architecture solutions applying GENE\_ARCH: An evolution-based generative design system. *Advanced Engineering Informatics*, 22(1), 59-70. doi:<http://dx.doi.org/10.1016/j.aei.2007.08.012>
- Caldas, L. G. (2001). *An evolution-based generative design system: using adaptation to shape architectural form*. Massachusetts: MIT. Retrieved October 30, 2013, from <http://18.7.29.232/handle/1721.1/8188#files-area>
- Calgary Heritage. (2007). *EnCana/Bow, Regis Hotel, York Hotel news*. Retrieved March 8, 2015, from Calgary Heritage: <http://www.calgaryheritage.org/phpbb/viewtopic.php?t=264>
- Choudhary, R., & Michalek, J. (2005). Design optimization in computer aided architectural design. *Proceedings of CAADRIA*, 149-158.
- Christopher, A. (1964). *Notes on the Synthesis of Form*. Cambridge: President and Fellows of Harvard College.
- Ciftcioglu, O., Bittermann, M., & Sariyildiz, S. (2007). *A Neural Fuzzy System for Soft Computing*. Retrieved from [http://www.bk.tudelft.nl/fileadmin/Faculteit/BK/Onderzoek/Projecten/Computational\\_Intelligent\\_Design/doc/205.pdf](http://www.bk.tudelft.nl/fileadmin/Faculteit/BK/Onderzoek/Projecten/Computational_Intelligent_Design/doc/205.pdf)
- Coello Coello, C., Lamont, G., & Van Veldhuizen, D. (2007). *Evolutionary algorithms for solving multi-objective (2nd Edition)*. New York: Springer Science+Business Media.
- Cross, N., & Maver, T. (1973). Computer Aids for Design Participation. *Architectural Design*, 53(5), 274.

- Cunha, A., Loyens, D., & van Hattum, F. (2011). Aesthetic design using multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization - Lecture Notes in Computer Science* (pp. 374-388). Berlin Heidelberg: Springer.
- Davis, D. (2009). *Evolving digital morphogenesis by means of biology and computer science*. Melbourne: RMIT.
- De Weck, O. (2004). Multiobjective Optimization: History and Promise. *Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, 2. Kanazawa.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, England, UK: John Wiley & Sons.
- DeLanda, M. (2002). Deleuze and the Use of the Genetic Algorithm in Architecture. *Architectural Design*, 71(7).
- DTZ Zadelhoff. (2014). *The Netherland, a national picture: Office and industrial property markets*. DTZ Zadelhoff.
- EC Harris. (2006). *Indicative building costs*. Paul Moore.
- Evins, R. (2013). A review of computational optimisation methods applied to sustainable building design. *Renewable and sustainable energy reviews*, 22, 230-245.  
doi:dx.doi.org/10.1016/j.rser.2013.02.004
- Fasoulaki, E. (2008). *Integrated Design: A Generative Multi-Performative Design Approach*. Cambridge: MIT .
- Foster + Partners. (2013). *Projects/ The Bow*. Retrieved February 2, 2015, from Foster + Partners: <http://www.fosterandpartners.com/projects/the-bow/>
- Frazer, J. (1995). *An Evolutionary Architecture*. London: Architectural Association. Retrieved October 30, 2013, from <http://www.aaschool.ac.uk/publications/ea/intro.html>
- Gerber, D., & Flager, F. (2011). *Teaching Design Optioneering: A Method for Multidisciplinary Design Optimization*. Stanford University, Civil and Environmental Engineering Dept. Stanford: Stanford University.

- Gerber, D., Lin, S.-H., Pan, B., & Solmaz, A. (2012). Design optioneering: multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm. *SimAUD 2012*, (pp. 23-30). Orlando, Florida.
- Gero, J. (1975). Architectural Optimization - A Review. *Engineering Optimization*, 1(3), 189-199. doi:10.1080/03052157508960586
- Gero, J. (1985). *Design Optimization* (Vol. 1). (J. Gero, Ed.) London, UK: Academic Press Inc.
- Haymaker, J., & Flager, F. (2009). *A Comparison of Multidisciplinary Design, Analysis and Optimization Process in the Building Construction and Aerospace*. Civil and Environmental Engineering Dept. Stanford: Stanford University.
- Hensel, M. (2008, March-April). Performance-Oriented Design Precursors and Potentials. *Architectural Design*, 78(2), pp. 48-53.
- Holland, J. (1975). *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press.
- HR Reit. (2012). *The Bow, Calgary*. Retrieved February 22, 2015, from HR Reit: <http://www.hr-reit.com/Properties/Profiled-Properties/The-Bow-Calgary.aspx>
- Janssen, P. (2006). A Generative Evolutionary Design Method. *Digital Creativity*, 17(1), 49-63. doi:10.1080/14626260600665736
- Kalay, Y. (2004). *Architecture's new media : principles, theories, and methods of computer-aided design*. USA: MIT Press.
- Keough, I., & Benjamin, D. (2010). Mult-objective optimization in architectural design. *SimAUD 2010*. San Diego: Society for Computer Simulation International. doi:10.1145/1878537.1878736
- Kohonen, J. (1999). *A brief comparison of simulated annealing and genetic algorithm approaches*. Helsinki: Department of Computer Science, University of Helsinki. Retrieved from <http://www.cs.helsinki.fi/u/kohonen/papers/gasa.html>

- Kolarevic, B. (2004, January). Back to the Future: Performative Architecture. *International Journal of Architectural Computing*, 2(1), 43-50. doi:10.1260/1478077041220205
- Kolarevic, B. (2005). Towards the performative in architecture. In B. Kolarevic, & A. Malkawi, *Performative architecture: beyond instrumentality* (pp. 205-213). New York: Spon Press.
- Koster, H. (2004). *Dynamic Daylighting Architecture: Basics, Systems, Projects*. Verlag: Springer.
- Lin, S.-H., & Gerber, D. (2013). Designing-in performance: towards cloud based simulation and multidisciplinary design solution space search. *SimAUD 2013*, (pp. 139-146). San Diego, California.
- Malkawi, A. (2005). Performance simulation: research and tools. In B. Kolarevic, & A. Malkawi, *Performative Architecture - Beyond Instrumentality* (pp. 87-95). New York: Spon Press.
- Malkawi, A., Srinivasan, R., Yi, Y., & Choudhary, R. (2003). Performance-based design evolution: the use of Genetic Algorithms and CFD. *Proceeding of the 8th International IBPSA conference* (pp. 793-798). Eindhoven: Building Simulation.
- Marsault, X. (2013). A multiobjective and interactive genetic algorithm to optimize the building form in early design stages. *13th Conference of International Building Performance Simulation Association* (pp. 809-816). Chambery, France: BS2013.
- Maver, T. (1987). Software tools for the technical evaluation of design alternatives. *CAAD Futurees Digital Proceedings*, (pp. 47-58). Eindhoven. Retrieved from [http://elpub.architecturez.net/system/files/pdf/0347.content\\_0.pdf](http://elpub.architecturez.net/system/files/pdf/0347.content_0.pdf)
- Maver, T. (2002). Predicting the Past, Remembering the Future. *SIGraDi Proceedings of the 6th Iberoamerican Congress of Digital Graphics*, (pp. 2-3). Caracas. Retrieved from [http://cuminCAD.scix.net/cgi-bin/works/Show?\\_id=c7e9&sort=DEFAULT&search=%2fseries%3a%22sigradi%22&hits=1655](http://cuminCAD.scix.net/cgi-bin/works/Show?_id=c7e9&sort=DEFAULT&search=%2fseries%3a%22sigradi%22&hits=1655)
- Michalek, J., & Papalambros, P. (2002). Interactive design optimization of architectural layouts. *Engineering Optimization*, 34(5), pp. 485-501.

- Oxman, R. (2006). Theory and design in the first digital age. *Design Studies*, 27(3), 229-265.  
doi:10.1016/j.destud.2005.11.002
- Oxman, R. (2008, January). Performance-based Design: Current Practices and Research Issues. *International Journal of Architectural Computing*, 6(1), 1-17.
- Radford, A., & Gero, J. (1988). *Design Optimization in Architecture, Building and Construction*. New York: Van Nostrand Reinhold.
- Read, H. (1951). *Aspects of form*. (L. Whyte, Ed.) Bloomington, Indiana, USA: Indiana University Press.
- Reinhart, C., & Breton, P.-F. (2009, July 1). Experimental Validation of Autodesk 3ds Max Design 2009 and Dayism 3.0. *LEUKOS*, 6(1), pp. 7-35.
- Rosenberg, R. (1967). *Simulation of genetic populations with biochemical properties*. Ann Arbor, Michigan, USA: University of Michigan.
- Roy, R., Hinduja, S., & Teti, R. (2008). Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals-Manufacturing Technology*, 57(2), 697-715.
- Sanguinetti, P., Bernal, M., El-Khaldi, M., & Erwin, M. (2010). Real-time design feedback: coupling performance-knowledge with design iteration for decision-making. *SimAUD 2010*, (pp. 65-72). Orlando, Florida.
- Shea, K., Aish, R., & Gourtovaia, M. (2003). Towards Integrated Performance-Based Generative Design Tools. *Digital Design: 21st eCAADe Conference Proceedings*, (pp. 553-560). Graz. Retrieved from  
[http://cumincad.architecture.net/system/files/pdf/ecaade03\\_553\\_149\\_shea.content.pdf](http://cumincad.architecture.net/system/files/pdf/ecaade03_553_149_shea.content.pdf)
- The City of Calgary. (2007). *The City of Calgary: Land Use Bylaw 1P2007 Part 10*. Retrieved February 2, 2015, from The City of Calgary:  
[http://www.calgary.ca/\\_layouts/cocis/DirectDownload.aspx?target=http%3a%2f%2fwww.calgary.ca%2fPDA%2fpd%2fDocuments%2fCalgary-Land-Use-bylaw-1P2007%2fpart10\\_of\\_1p2007.pdf&noredirect=1&sf=1](http://www.calgary.ca/_layouts/cocis/DirectDownload.aspx?target=http%3a%2f%2fwww.calgary.ca%2fPDA%2fpd%2fDocuments%2fCalgary-Land-Use-bylaw-1P2007%2fpart10_of_1p2007.pdf&noredirect=1&sf=1)

- Turin, M., von Buelow, P., & Stouffs, R. (2011, October). Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. *Advanced Engineering Informatics*, 25(4), pp. 656-675.  
doi:10.1016/j.aei.2011.07.009
- University of Strathclyde. (n.d.). *History*. Retrieved from University of Strathclyde:  
<http://www.strath.ac.uk/architecture/departments/history/>
- Vierlinger, R. (2013). Octopus Manual. Retrieved from  
<http://www.grasshopper3d.com/group/octopus?groupUrl=octopus&id=2985220%3AGroup%3A742529&page=3>
- Vierlinger, R., & Bollinger, K. (2014). Accommodating Change in Parametric Design. *ACADIA 2014 Design Agency Proceedings* (pp. 609-618). Los Angeles: ACADIA & Riverside Architectural Press.
- Vierlinger, R., & Hofmann, A. (2013). Framework for Flexible Search and Optimization in Parametric Design. *Rethinking Prototyping: Proceedings of the Design Modelling Symposium Berlin 2013*. Berlin.
- Von Buelow, P. (2011). Genetically Enhanced Parametric Design for Performance Optimization. *Proceedings of the 7th Seminar of the IASS Structural Morphology Group International*. London.
- Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the strength Pareto Evolutionary Algorithm*. Swiss Federal Institute of Technology (ETH) Zurich, Zurich. Retrieved from <http://www.kddresearch.org/Courses/Spring-2007/CIS830/Handouts/P8.pdf>

## Appendix A

Note: Details can be seen by zooming when viewing as a PDF.

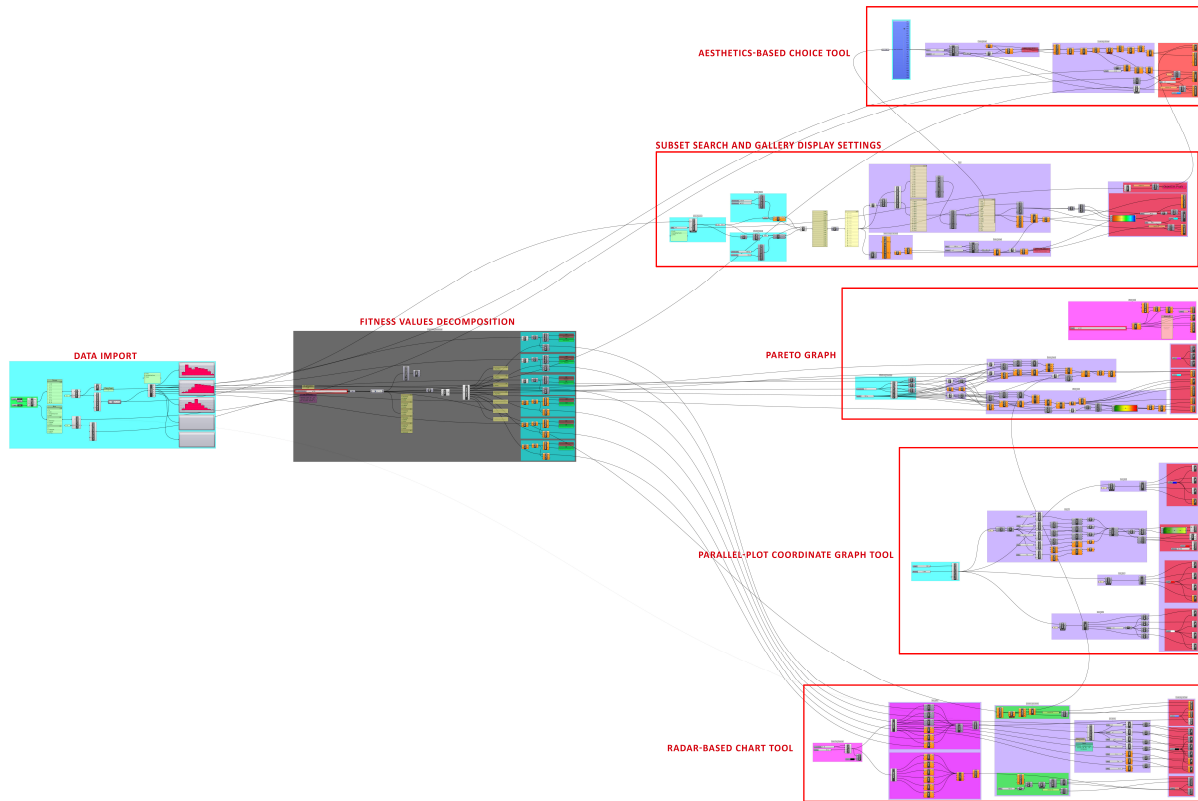


Figure 7-1. Creative Optimization Tools.

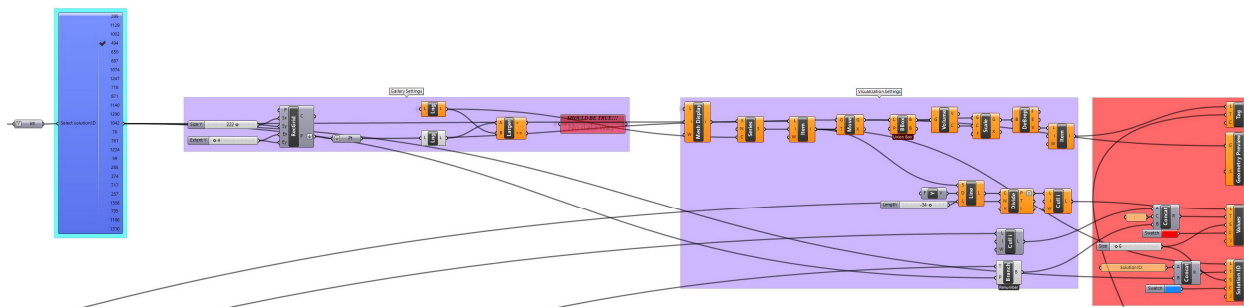


Figure 7-2. Form-based choice tool.



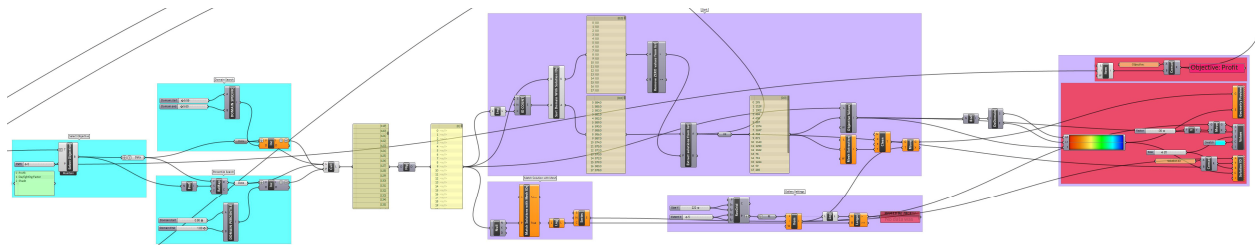


Figure 7-3. Subset search and gallery display settings.

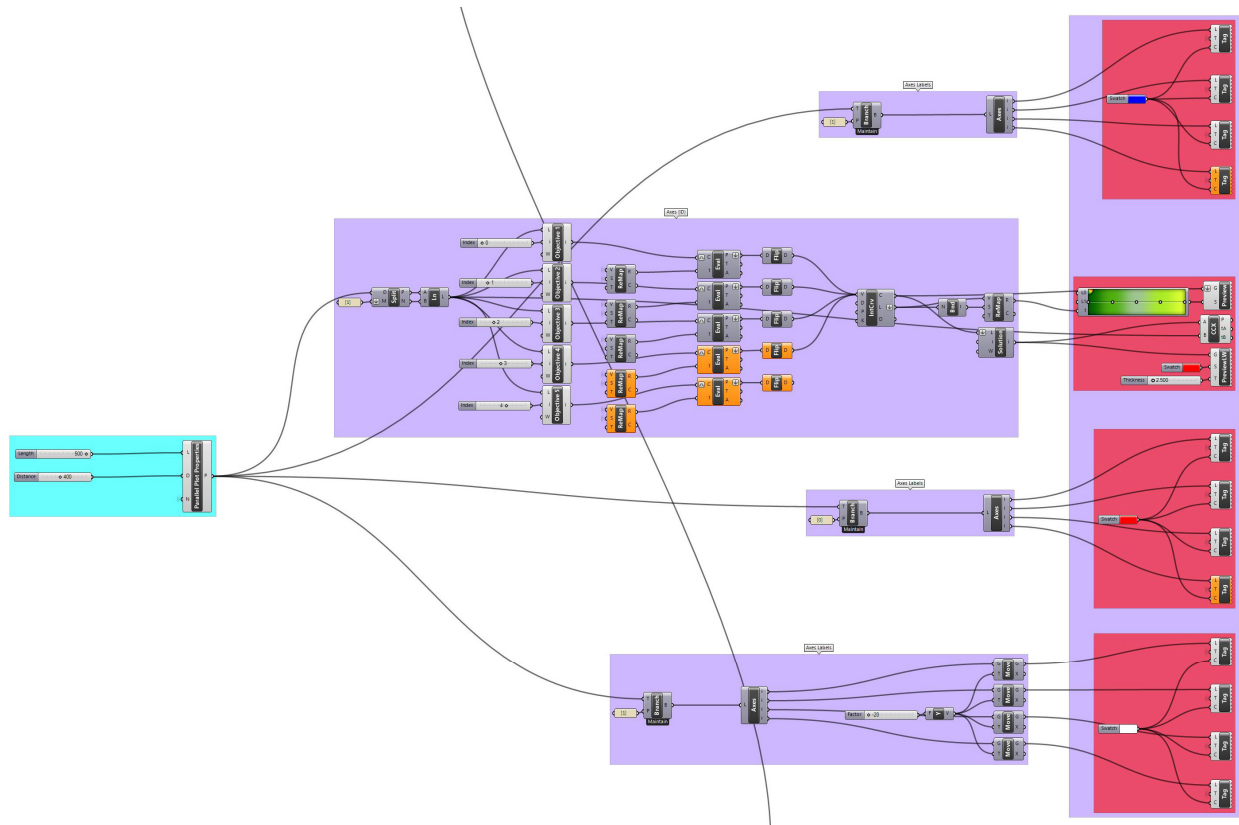


Figure 7-4. Parallel-plot coordinate graph tool.

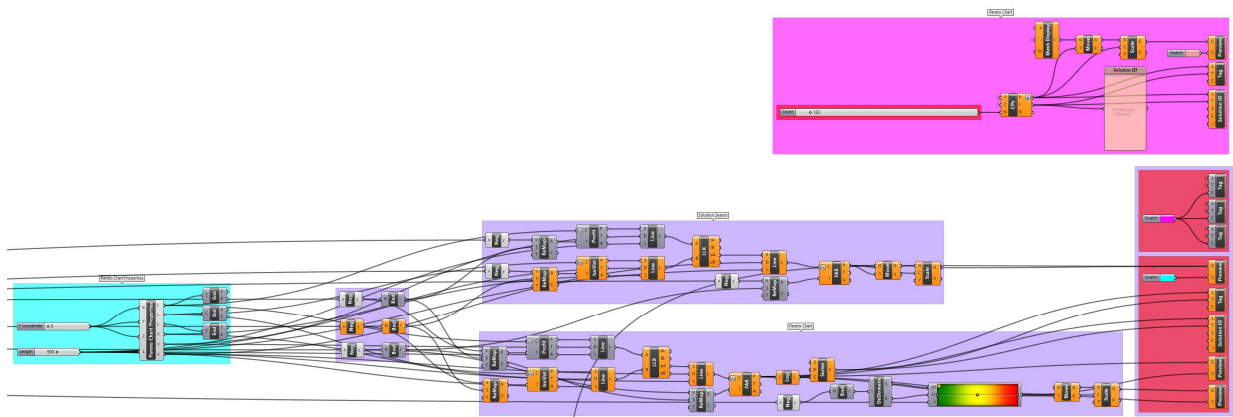


Figure 7-5. Pareto graph tool.

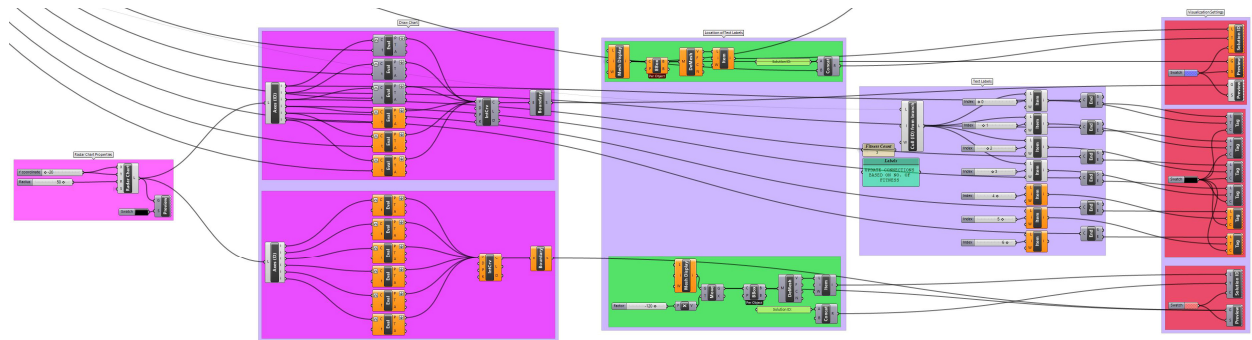


Figure 7-6. Radar-based chart tool.

## Appendix B

Note: Details can be seen by zooming when viewing as a PDF.

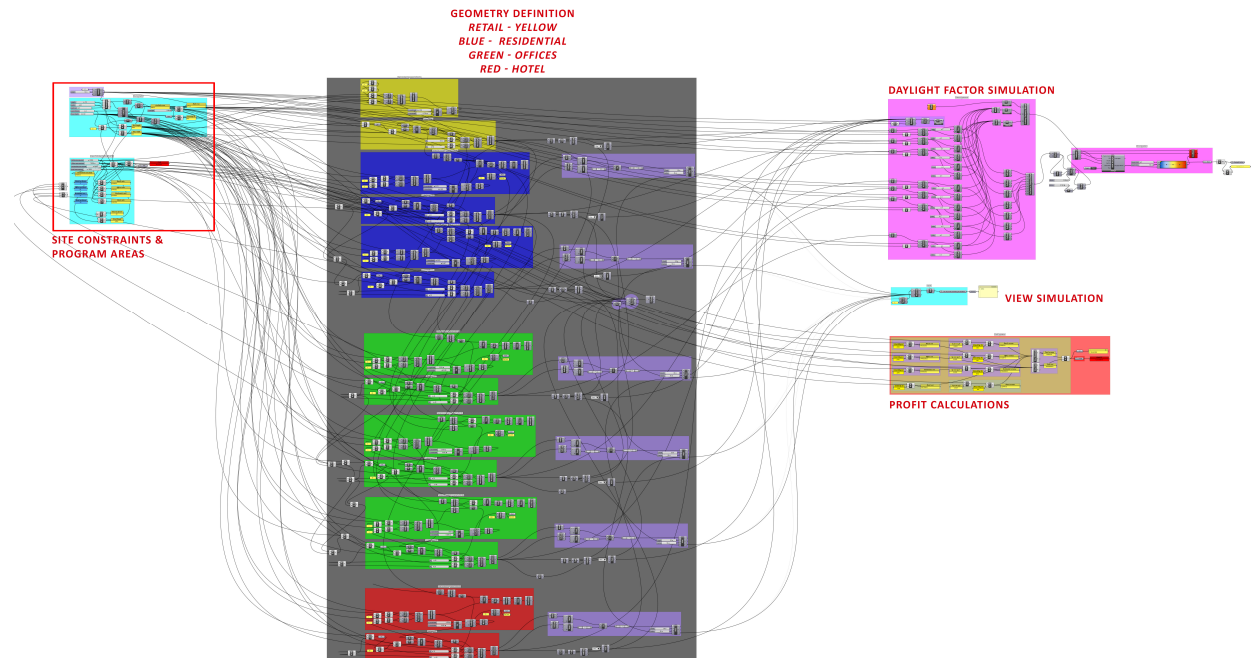


Figure 8-1. De Rotterdam parametric replica GH definition.

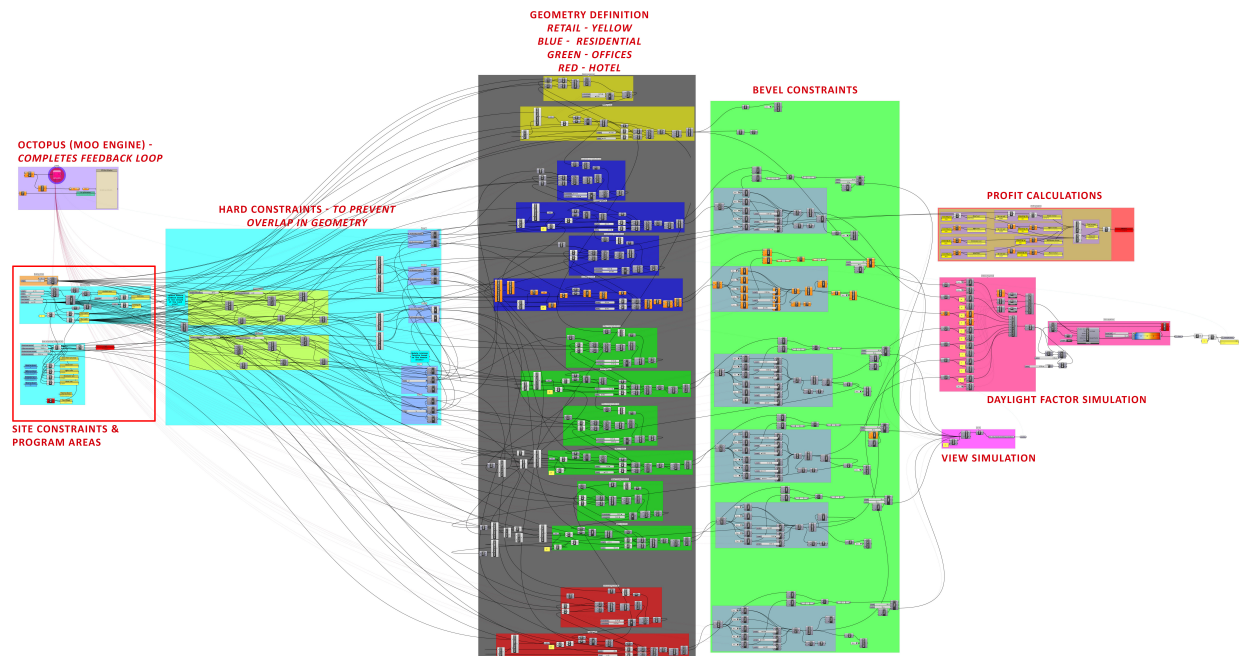


Figure 8-2. De Rotterdam (bevel iteration) GH definition.

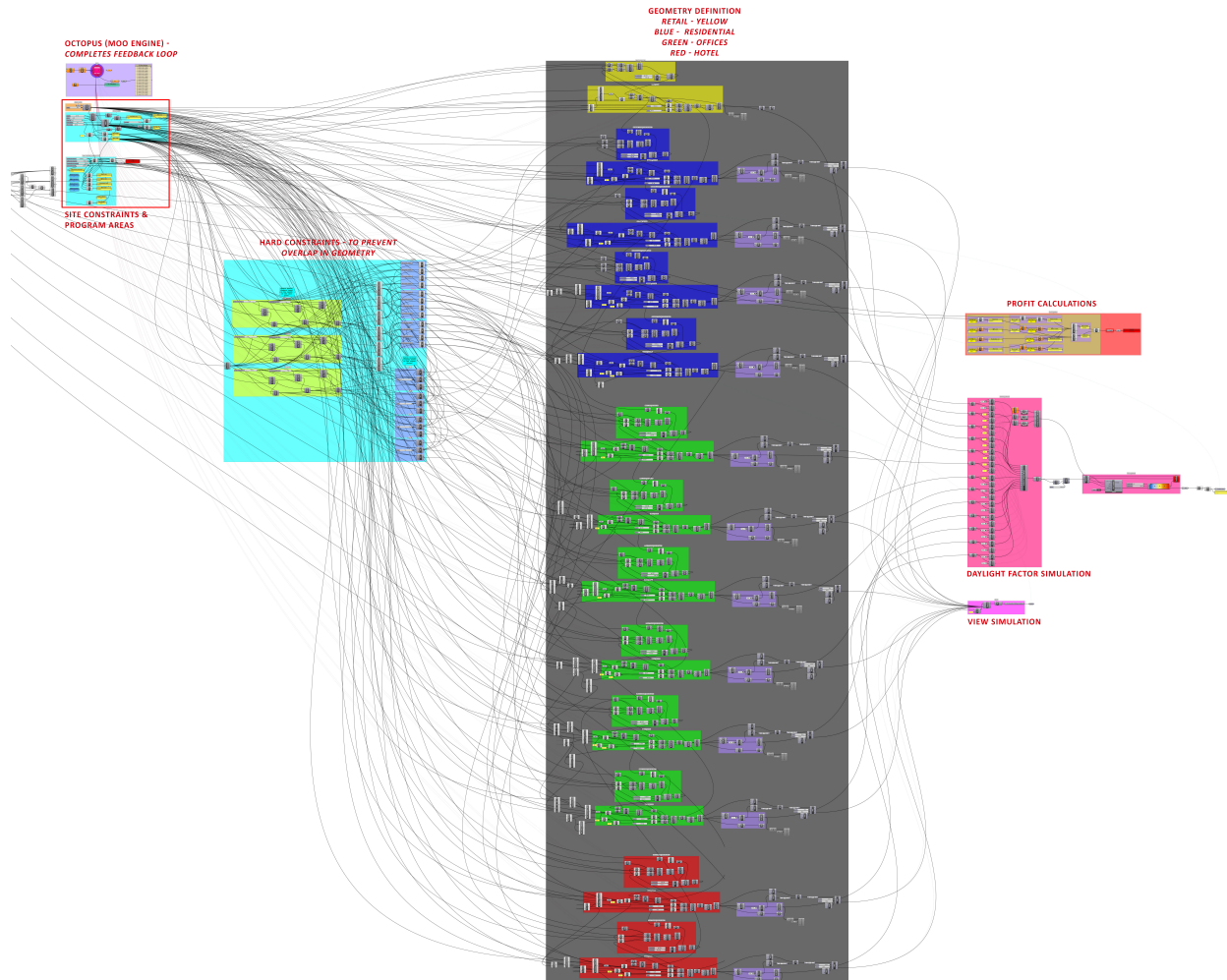


Figure 8-3. De Rotterdam (stacked block iteration) GH definition.

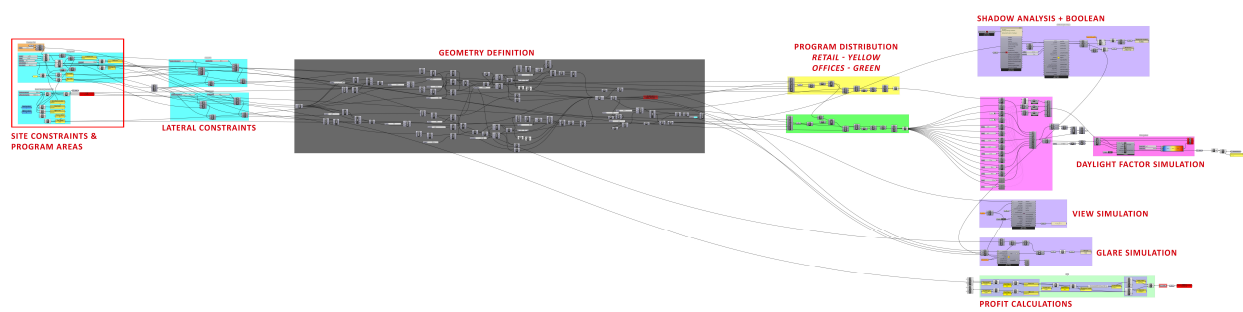


Figure 8-4. The Bow Tower parametric replica.

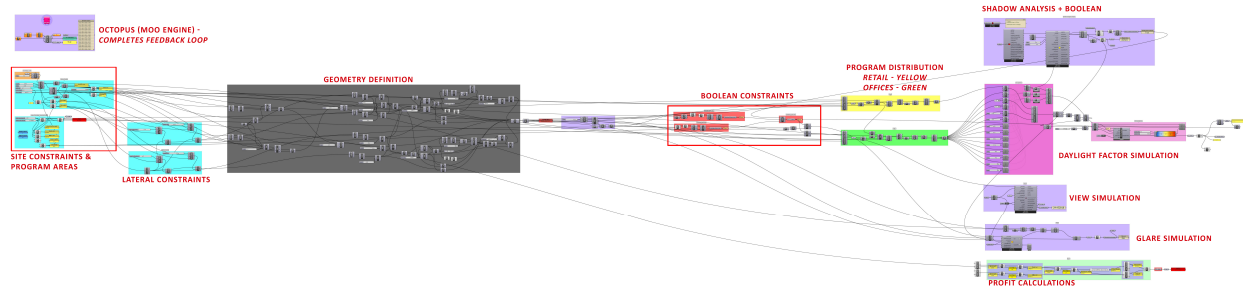


Figure 8-5. The Bow Tower original geometry (1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup> optimization runs).

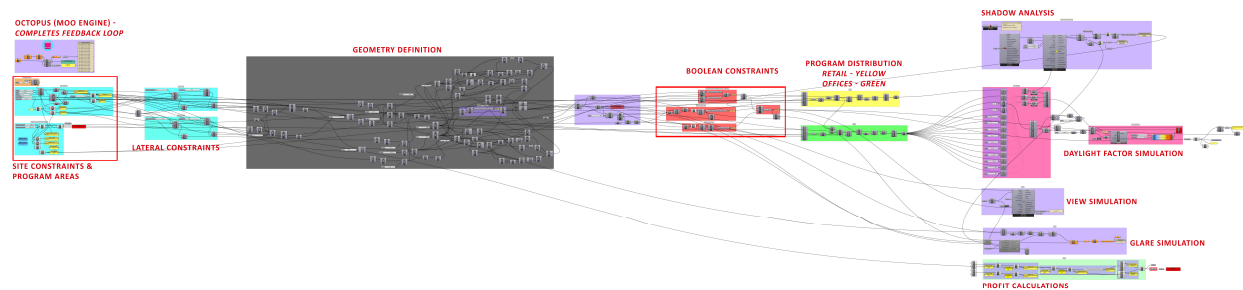


Figure 8-6. The Bow Tower modified geometry (4<sup>th</sup>, 5<sup>th</sup>, 6<sup>th</sup>, 7<sup>th</sup> optimization runs).

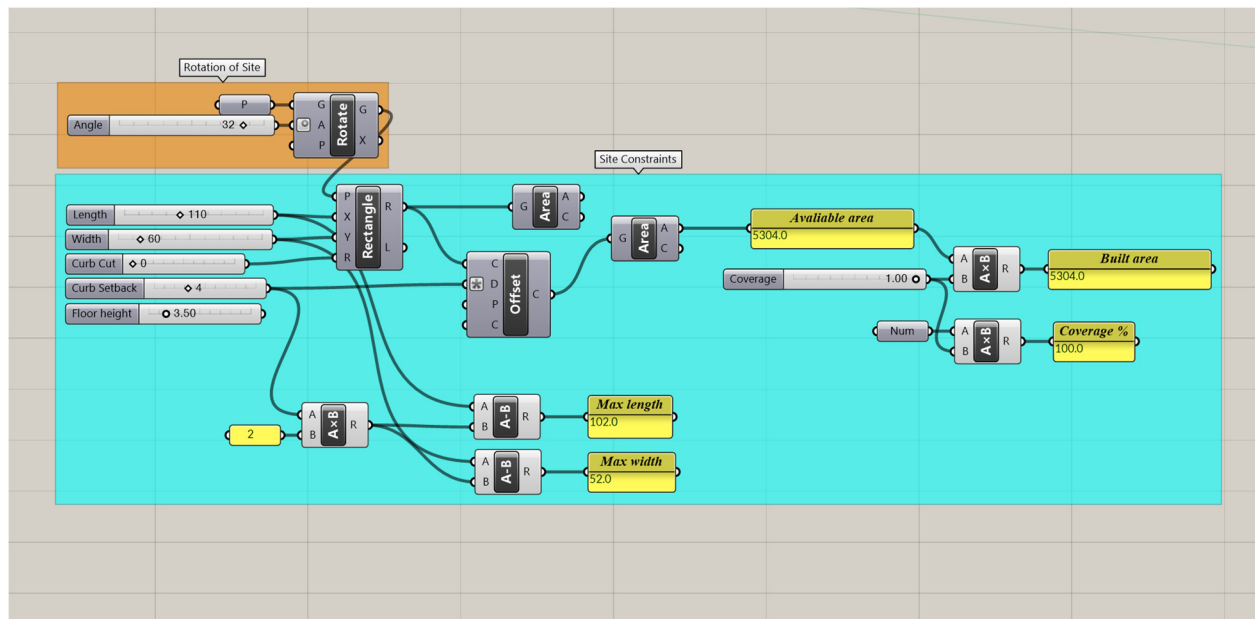


Figure 8-7. Site constraints.



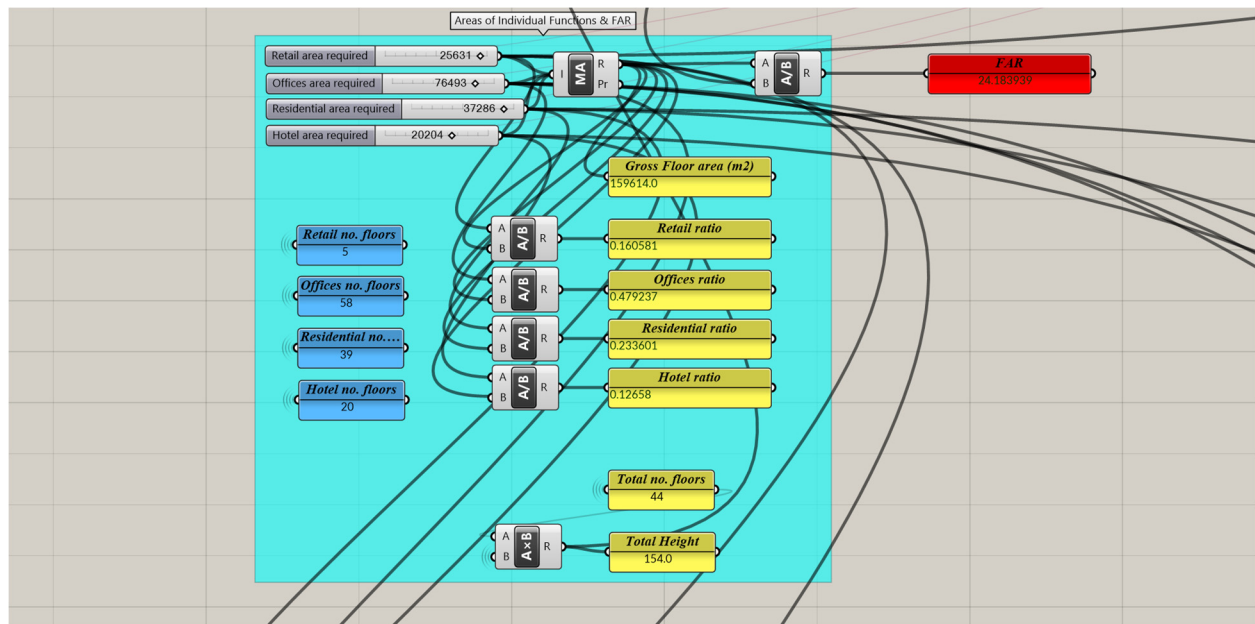


Figure 8-8. Program definition areas.

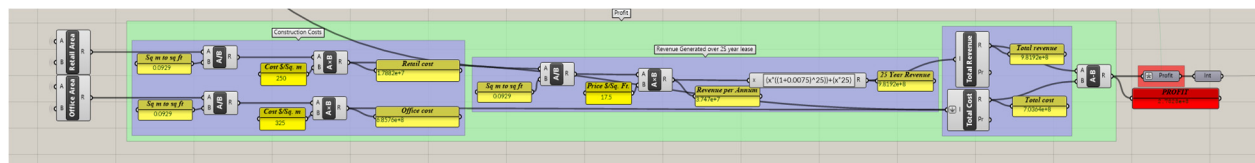


Figure 8-9. Financial performance: profit calculation (the Bow Tower)

## Appendix C

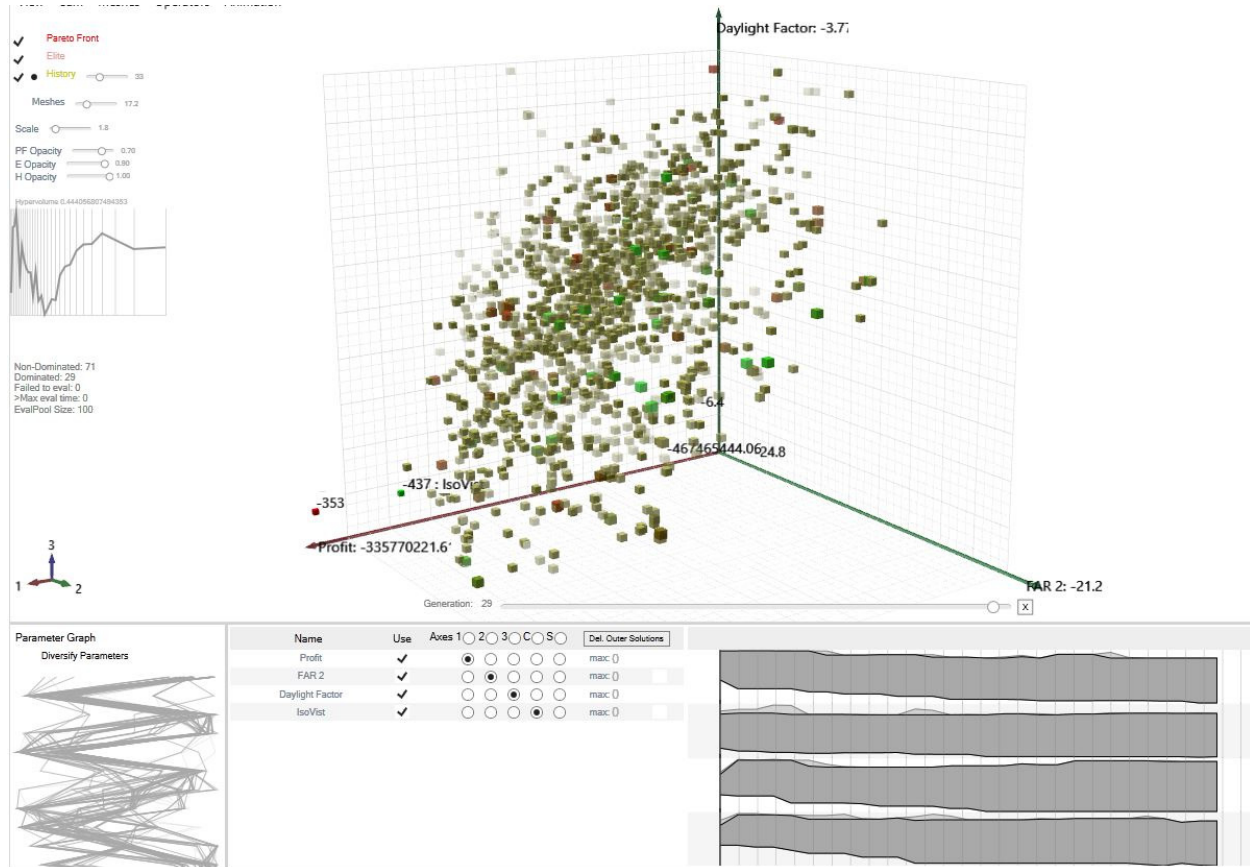


Figure 9-1. De Rotterdam (bevel iteration) Octopus results.

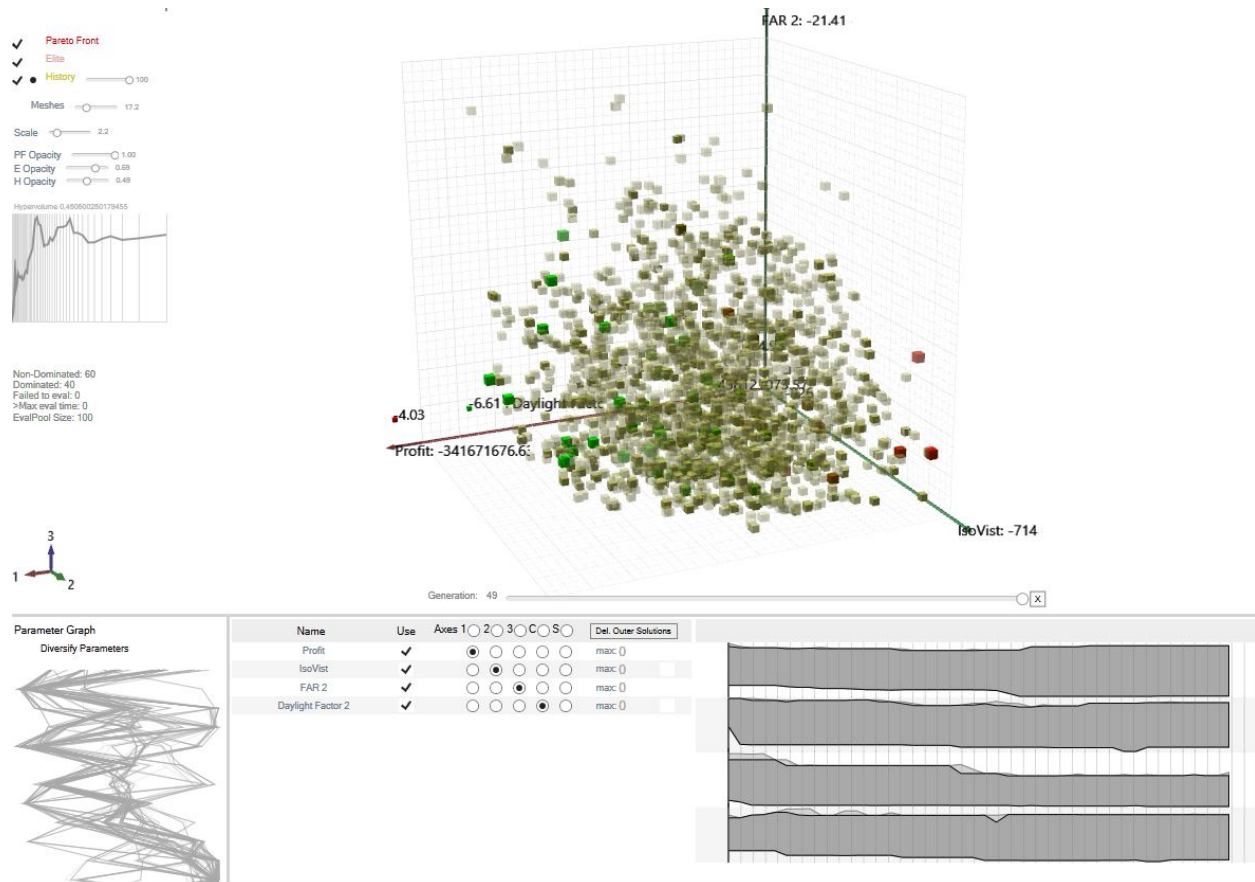


Figure 9-2. De Rotterdam (stacked block iteration) Octopus results.



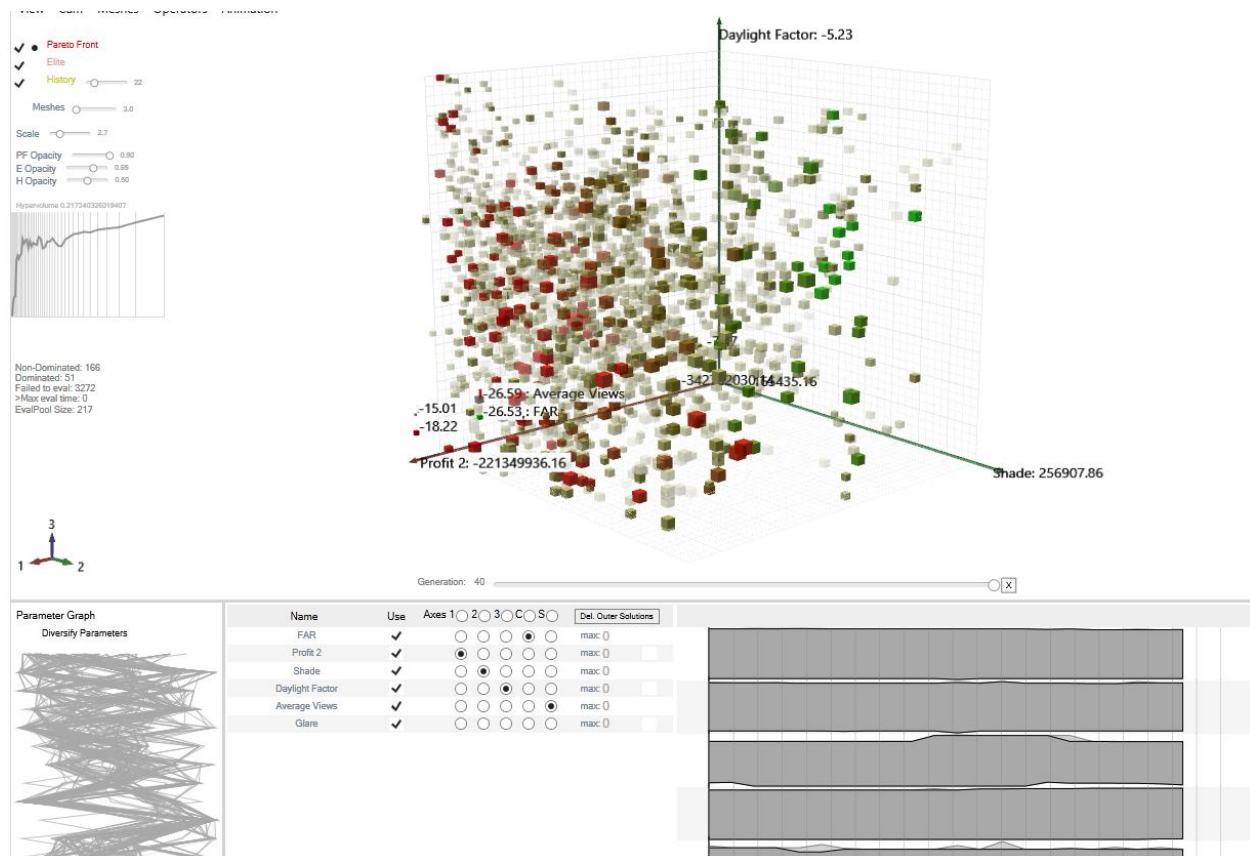


Figure 9-3. Bow Tower 1<sup>st</sup> optimization run (6 objectives) Octopus results.

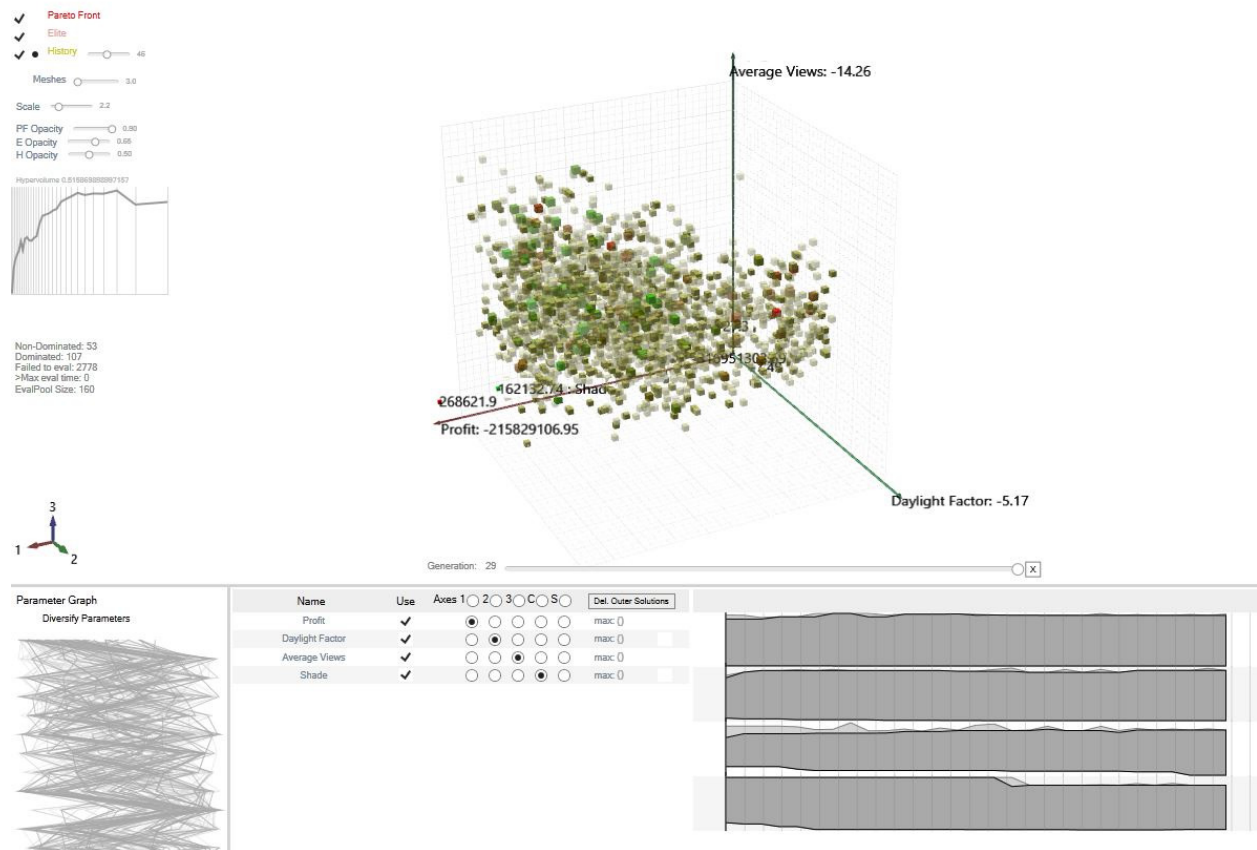


Figure 9-4. Bow Tower 2<sup>nd</sup> optimization run (4 objectives) Octopus results.

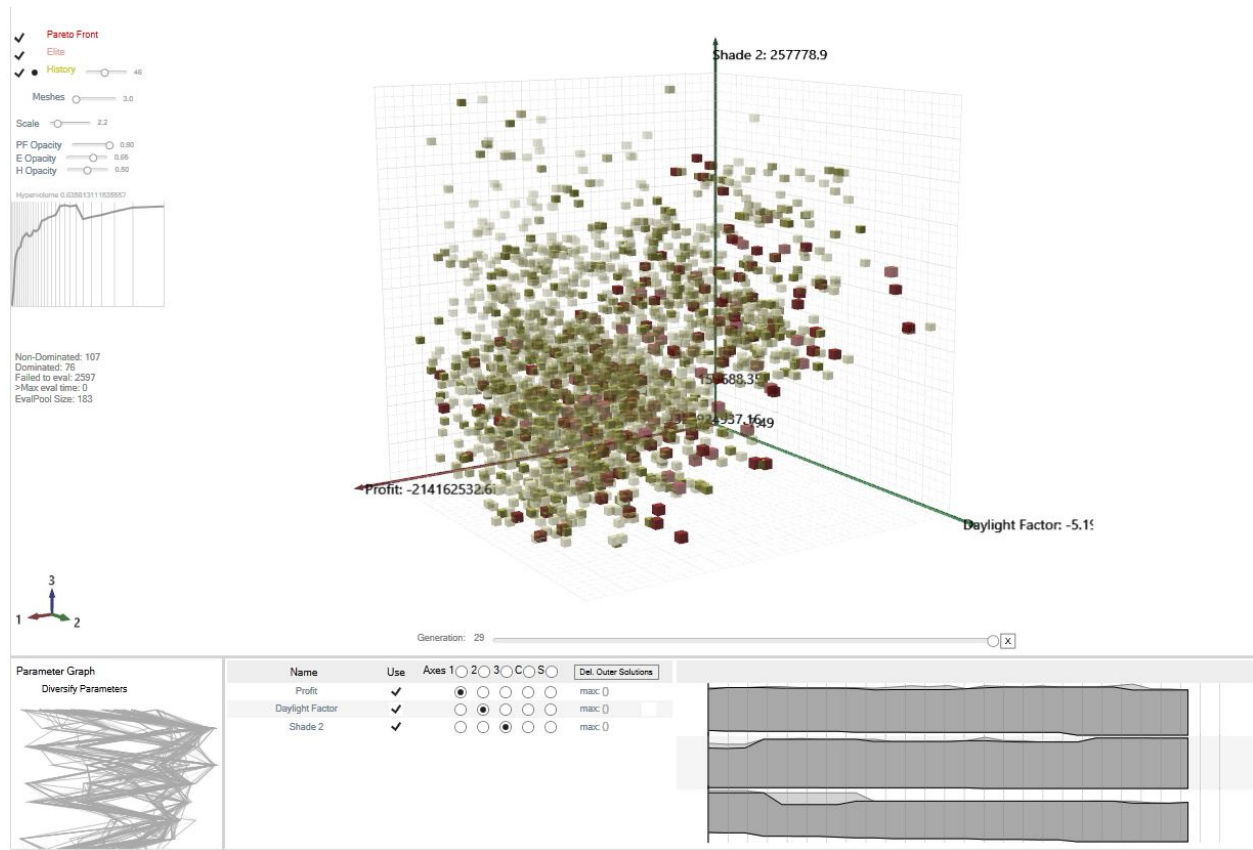


Figure 9-5. Bow Tower 3<sup>rd</sup> optimization run (3 objectives) Octopus results.

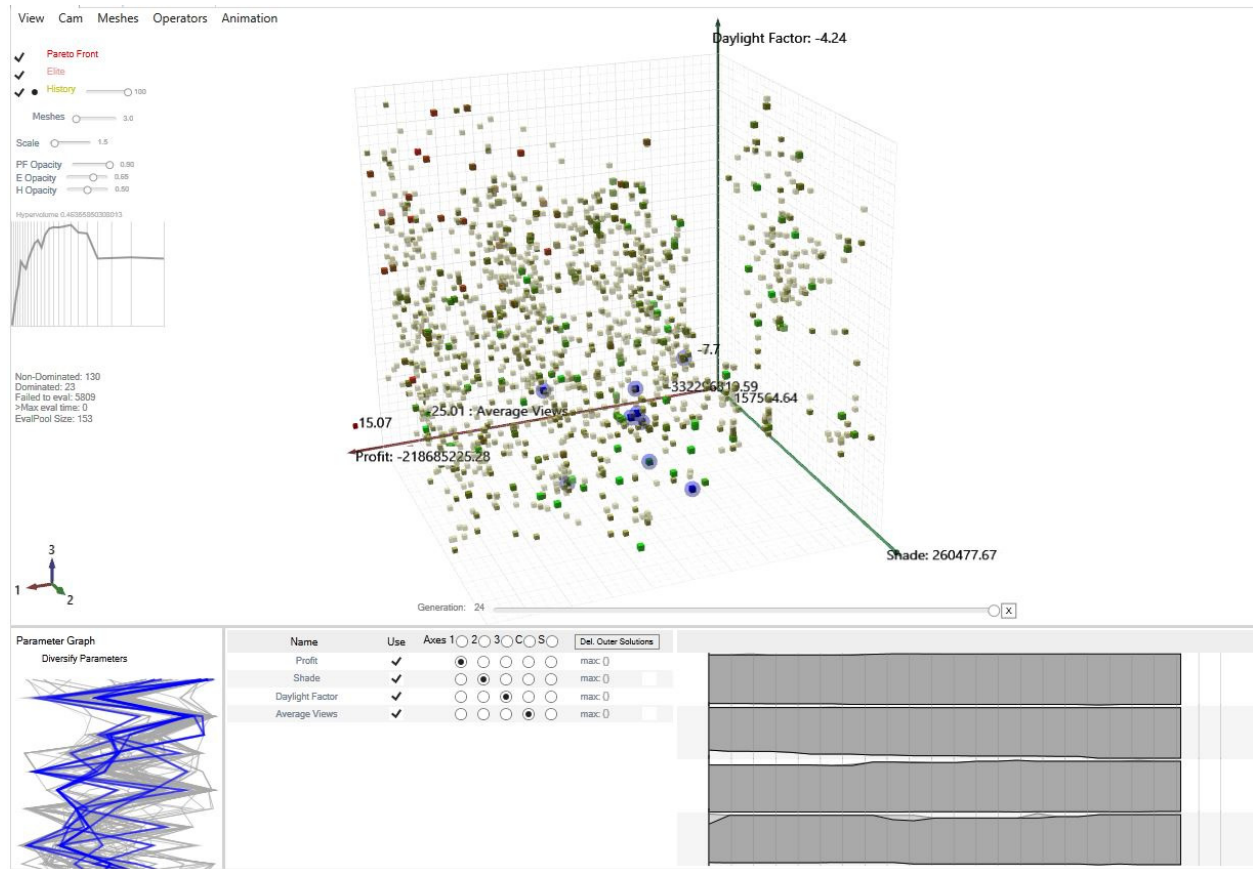


Figure 9-6. Bow Tower 4<sup>th</sup> optimization run (4 objectives) Octopus results.

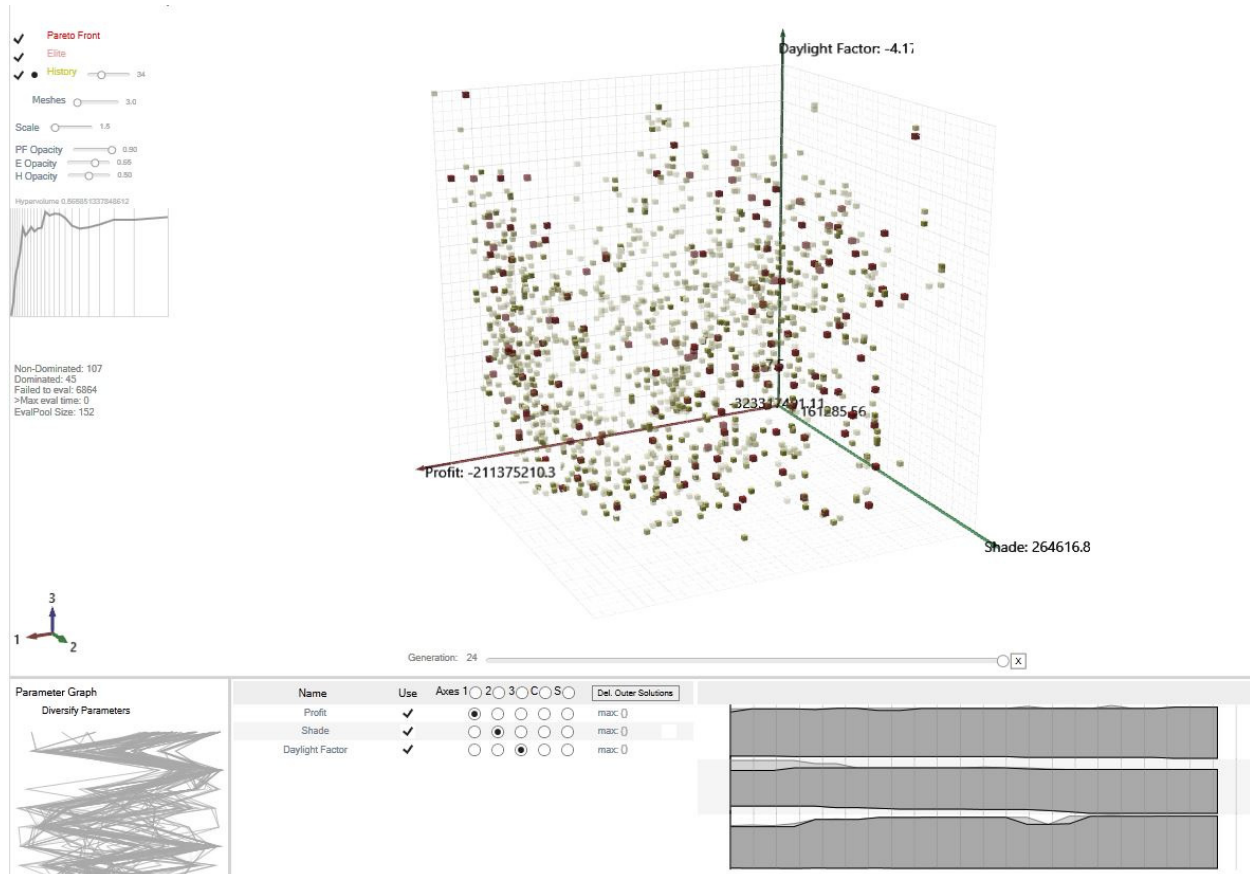


Figure 9-7. Bow Tower 5<sup>th</sup> optimization run (3 objectives) Octopus results.



## Appendix D

Table 10-1 De Rotterdam 1<sup>st</sup> optimization run (bevel iteration) results.

	Lower Value	Upper Value	Average
FAR	21.2	24.80	23
Profit	\$335,770,000	\$467,470,000	\$401,620,000
Daylight Factor	3.77216%	6.40%	5.085%
Views	353	437	395

Table 10-2. De Rotterdam 2<sup>nd</sup> optimization run (stacked block iteration) results.

	Lower Value	Upper Value	Average
FAR	21.1	25	23.05
Profit	\$341,670,000	\$458,120,000	\$399,895,000
Daylight Factor	4.03%	6.61%	5.32%
Views	353	413	383

Table 10-3. Bow Tower 1<sup>st</sup> optimization run (6 objectives) results.

	Lower Value	Upper Value	Average
Profit	\$215,597,472	\$347,494,661	\$281,546,067
Daylight Factor	5.19%	7.50%	6.35%
Shaded Area	160707 m <sup>2</sup>	268845 m <sup>2</sup>	214776 m <sup>2</sup>
Views	13%	27%	20%
Glare	160	553	357
FAR	17.99	26.7	22.35

Table 10-4. Bow Tower 2<sup>nd</sup> optimization run (4 objectives) results.

	Lower Value	Upper Value	Average
--	-------------	-------------	---------

Profit	\$215,829,107	\$316,951,304	\$266,390,206
Daylight Factor	5.17%	7.51%	6.34%
Shaded Area	162133 m <sup>2</sup>	268622 m <sup>2</sup>	215378 m <sup>2</sup>
Views	14%	27%	20.5%

Table 10-5. Bow Tower 3<sup>rd</sup> optimization run (3 objectives) results.

	Lower Value	Upper Value	Average
Profit	\$214,162,533	\$325,924,937	\$270,043,735
Daylight Factor	5.19%	7.5%	6.35%
Shaded Area	159688 m <sup>2</sup>	257779 m <sup>2</sup>	208734 m <sup>2</sup>

Table 10-6. Bow Tower 4<sup>th</sup> optimization run (4 objectives) results.

	Lower Value	Upper Value	Average
Profit	\$218,685,225	\$332,296,320	\$275,490,773
Daylight Factor	4.24%	7.7%	5.97%
Shaded Area	157565 m <sup>2</sup>	260478 m <sup>2</sup>	209022 m <sup>2</sup>
Views	15%	25%	20%

Table 10-7. Bow Tower 5<sup>th</sup> optimization run (3 objectives) results.

	Lower Value	Upper Value	Average
Profit	\$211,375,210	\$323,317,491	\$267,346,351
Daylight Factor	4.169%	7.5%	5.84%
Shaded Area	161286 m <sup>2</sup>	264617 m <sup>2</sup>	212952 m <sup>2</sup>

Table 10-8. Bow Tower 6<sup>th</sup> optimization run (2 objectives) results.

	Lower Value	Upper Value	Average
Profit	\$223,089,832	\$342,124,326	\$282,607,079
Daylight Factor	4.21%	7.50%	5.85%



Table 10-9. Bow Tower 7<sup>th</sup> optimization run (1 objective) results.

	Lower Value	Upper Value	Average
Profit	\$223,766,987	\$342,190,136	\$282,978,561