

## **Optimization in Architecture**

**Catarina Garcia Belém**

Thesis to obtain the Master of Science Degree in  
**Information Systems and Computer Engineering**

Supervisor: Prof. António Menezes Leitão

**May 2019**



# Acknowledgments

I would like to express my respect and gratitude to my supervisor and friend Dr. António Menezes Leitão. He proposed an interesting theme, which proved to be intriguing and challenging. His efforts to arrange research grants and to supply better computational resources were inspiring and encouraged me to fight the difficulties found along the way. His constant support, preoccupation and first-class guidance were invaluable through this thesis. Thanks for everything, especially for encouraging me to pursue my dreams and for providing me with the flexibility and free-will to tackle this theme as something that I would be proud of.

I would like to thank the members of the research group oriented by my supervisor, the Grupo de Arquitetura e Computação (GAC), for their support and valuable ideas and discussions which undoubtedly improved the practicality of this work - especially, Inês Caetano, Inês Pereira, Renata Castelo Branco, Guilherme Ilunga, and Luís Silveira Santos.

I would also like to thank the Department of Computer Science and Engineering at Instituto Superior Técnico, Universidade de Lisboa for providing me with the foundations for completing this work, as well as for the opportunities to lecture as a teaching assistant during my MSc Thesis. I would also like to thank the Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID) for the financial support provided to me in the form of Bachelor's Research Grants.

Moreover, I am also grateful to the staff and teachers of the Computer Engineering and Information Systems course for their friendship, their availability to discuss different subjects, and for providing an interesting working environment.

To all my friends whose support was invaluable during this period and which encouraged me to constantly push my limits when the task felt too large, I thank you deeply from my heart - especially, Carolina Pereira, Cristiana Tiago, Diogo Magalhães, Filipe Magalhães, Gonçalo Rodrigues, Guilherme Ilunga, Nuno Afonso, Pedro Simão, Rita Amaro, and Telma Correia.

Last but not least, I would like to thank my parents for their friendship, encouragement and caring over all these years, for always being there for me through thick and thin and without whom this project would not be possible. I would also like to thank my sister, brother, and sister-in-law, for their understanding, support and preoccupation throughout this year.

To each and every one of you – Thank you.

# Publications

The development of this thesis resulted in several scientific contributions exploring different perspectives of optimization problems:

1. Caetano, I., Ilunga, G., **Belém, C.**, Aguiar, R., Feist, S., Bastos, F., and Leitão, A. (2018). Case Studies on the Integration of Algorithmic Design Processes in Traditional Design Workflows. Proceedings of the 23rd International Conference of the Association for CAADRIA, 1(Giedion 1941), 111–120.
2. **Belém, C.**, and Leitão, A. (2018). From Design to Optimized Design An algorithmic-based approach. Proceedings of the 36th eCAADe Conference - Volume 2, Lodz University of Technology, Poland, 549-558



# **Abstract**

## **Keywords**

Algorithmic Design; Black-Box Optimization; Machine Learning; Surrogate-based Modelling.





# Resumo

## Palavras Chave

Design Algorítmico; Otimização de caixa-preta; Modelos baseados em aproximações; Aprendizagem Máquina.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	From design to Optimized design . . . . .	4
1.1.1	Building Performance Optimization . . . . .	5
1.1.2	Algorithmic Design . . . . .	6
1.1.3	Algorithmic Analysis . . . . .	8
1.1.4	Architectural Optimization Workflow . . . . .	10
1.2	Goals . . . . .	11
1.3	Organization of the Document . . . . .	11
<b>2</b>	<b>Background</b>	<b>13</b>
2.1	Single-Objective Optimization . . . . .	15
2.1.1	Derivative-Free Optimization . . . . .	15
2.1.2	Optimization Tools in Architecture . . . . .	15
2.1.2.A	Galapagos . . . . .	15
2.1.2.B	Goat . . . . .	15
2.2	Multi-Objective Optimization . . . . .	15
2.2.1	Experimental Approach . . . . .	16
2.2.2	Priori Articulation Approach . . . . .	16
2.2.3	Pareto-Based Approach . . . . .	16
2.2.4	Metrics for Multi-Objective Optimization . . . . .	16
2.2.5	Optimization Tools in Architecture . . . . .	16
2.2.5.A	Octopus . . . . .	16
2.2.5.B	Opossum . . . . .	16
2.2.5.C	Optimo . . . . .	16
<b>3</b>	<b>Solution</b>	<b>17</b>
3.1	Architecture Overview . . . . .	19
3.2	Architecture Design Requirements . . . . .	19
3.2.1	Problem Modelling . . . . .	19

3.2.2	Simple Solver . . . . .	19
3.2.3	Meta Solver . . . . .	19
3.3	Architecture Design Implementation . . . . .	19
3.3.1	Problem Modelling . . . . .	19
3.3.2	Simple Solver . . . . .	19
3.3.3	Meta Solver . . . . .	19
<b>4</b>	<b>Evaluation</b>	<b>21</b>
4.1	Qualitative Evaluation . . . . .	23
4.2	Quantitative of Applications . . . . .	23
4.2.1	Ericeira House: Solarium . . . . .	23
4.2.2	Black Pavilion: Arts Exhibit . . . . .	23
4.2.2.A	Skylights Optimization . . . . .	23
4.2.2.B	Arc-shaped Space Frame Optimization . . . . .	23
<b>5</b>	<b>Conclusion</b>	<b>25</b>
5.1	Conclusions . . . . .	27
5.2	System Limitations and Future Work . . . . .	27
5.2.1	Optimization Algorithms . . . . .	27
5.2.2	ML models . . . . .	27
5.2.3	Constrained Optimization . . . . .	27

# List of Figures

1.1	General views of Traditional Design Approaches . . . . .	5
1.2	General view of the Algorithmic Design Approach . . . . .	6
1.3	Design variations of the Astana's National Library . . . . .	7
1.4	General view of the Algorithmic Design and Analysis design approach . . . . .	9
1.5	3D and Analytical models of the Astana's National Library . . . . .	9



## **List of Tables**

## **List of Algorithms**





# Listings



# Acronyms

<b>AD</b>	Algorithmic Design
<b>AA</b>	Algorithmic Analysis
<b>BIM</b>	Building Information Modelling
<b>BPO</b>	Building Performance Optimization
<b>BPS</b>	Building Performance Simulation
<b>CAD</b>	Computer-Aided Design
<b>HVAC</b>	Heating, Ventilation, and Air Conditioning
<b>MOO</b>	Multi-Objective Optimization



# 1

## Introduction

### Contents

---

1.1	From design to Optimized design . . . . .	4
1.2	Goals . . . . .	11
1.3	Organization of the Document . . . . .	11

---



The act of making something as fully perfect, functional or effective as possible is a behavior that is constantly sought by us, Humans, in a process known as optimization [1]. Intuitively, through optimization one aims to improve a system in terms of different quantitative measurable aspects. Although usually striving to fully optimize these systems, i.e., to obtain *perfect* systems, it is often the case, that finding a better one or a near-optimal system suffices.

Generally, optimization processes are composed of two main parts: (1) the model of the system to be optimized and (2) the algorithm responsible for finding the optima. Conceptually, the model is a description of the system that is defined in terms of: a set of the system's characteristics, known as variables or unknowns, a set of quantitative measures of the system's performance, referred to as objectives or criteria, and, optionally, by a set of conditions that have to be satisfied to guarantee the system's feasibility, i.e., the system's constraints [2]. The objectives are usually functions of the variables being defined. Subsequent to the model definition, the obtained description can be interpreted as an optimization problem for which the optimal solutions are to be found, thus entering in the second part of an optimization process. In the second part, one executes an optimization algorithm, which encloses a description of the steps necessary to attain optimal solutions, which according to the user's intentions can be the maximization or minimization of the model's objectives.

Depending on the model representation, one is able to classify optimization problems differently with respect, for example, to its objective functions, variables, and determinacy. Due to their relevance in the developed work, in the next two paragraphs, we describe four different optimization classifications. However, we refer the interested reader to [2] for a more detailed and complete description of the different classifications.

One important classification is regarding the cardinality of the solutions sought by optimization processes, thus yielding the continuous and discrete optimization categories. In the former, optimal solutions lie in a potentially infinite set of candidate solutions, whereas in the latter, optimal solutions lie in a finite set. Optimization problems can also be classified as constrained or unconstrained, depending on whether the models explicitly define constraints or not. Moreover, optimization can also be distinguished in terms of the aim of the search that is performed, particularly, whether it is global or local. In local optimization, the search process strives to find a solution that is locally optimal, i.e., for which its value is better than all other points in its vicinity. The points that satisfy the previous property are known as local optima. On the other hand, there are optimization processes that strive to find the globally optimal solutions, i.e., the best of all the local optima.

Optimization is frequently used to address problems involving more than one objective. It is often the case that people face daily decisions involving two or more conflicting objectives, either to effectively manage resources, or just to ponder several factors associated with certain decisions. As an example, consider the decision of how to commute to work: either by car, or by bus. Indeed, in this case, the



optimal solution is to take the transport that minimizes the cost and the time spent in the commutation. When considering the two transports, one must consider the time-cost trade-off corresponding to the two different transports: (a) taking the car will incur in more costs but in less time spent in the trip, whereas (b) taking the bus incur in fewer costs but more time spent. This example belongs to the subset of Multi-Objective Optimization (MOO) problems which consider the optimization of more than one objective function.

In addition to day-to-day life decisions, optimization can also be used with different decision and analysis purposes. As a result, several areas, including economy, science, engineering, among others apply optimization as auxiliary tool to maximize the efficiency of the decisions involved. Particularly, architecture is one of the areas where the potential of optimization becomes more visible. The architectural practice can benefit from optimization to reduce the building sector's economic and ecological footprint through the finding of more efficient buildings variants before their construction. Given its importance to the world's sustainability and economy, this thesis focus on the application of optimization processes to enhance the architectural practice, providing, in the following sections, an overview of the involvement and the evolution of such processes in architecture. We end this chapter by highlighting our research goals and by outlining the upcoming document's structure.

## 1.1 From design to Optimized design

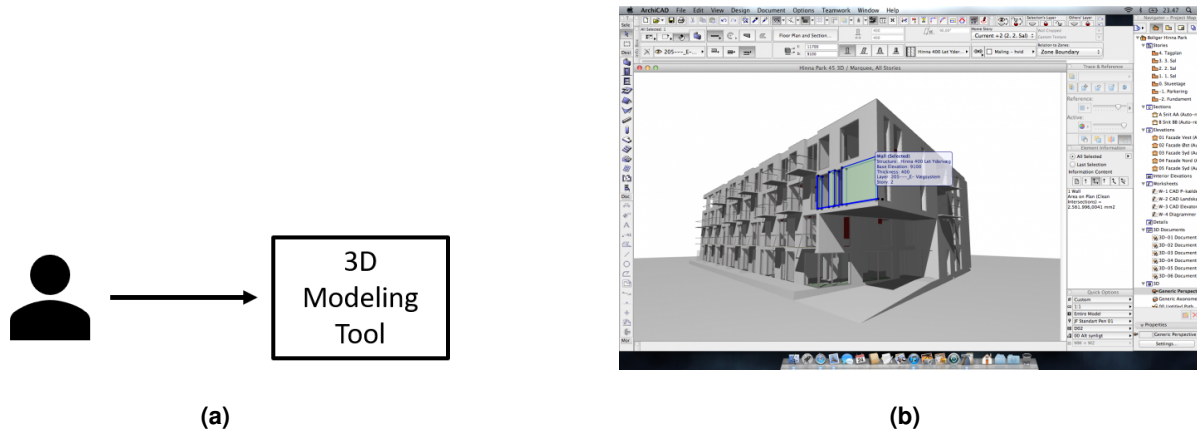


In the architectural practice, optimization has been gaining relevance for the past few years [3], especially due to the impact of building construction and building maintenance in the economy and environment. For this reason, designers are shifting from a pure aesthetically-based to performance-based design, where buildings are being optimized to achieve the best possible values regarding different aspects of their design, such as thermal comfort, energy consumption, lighting comfort, structural behavior, cost, among others.

This has only been possible due to the technological improvements in the architectural practice over the last few decades. The adoption of computer science techniques was responsible for the dissemination of digital modelling tools, which allowed the more accurate and efficient design of highly complex buildings. These tools enabled a shift from traditional paper-based approaches to more computerized ones, such as Computer-Aided Design (CAD) and Building Information Modelling (BIM) applications, where changes to designs are trivial and do not require manually erasing and redrawing parts of the original design [4]. Figure 1.1 illustrates the general view of this computer-aided design process, as well as an example of a 3D modeling tool. The architect interacts directly with these modeling tools to incrementally concretize his design ideas.

Shortly after, the development of computer-based simulation tools allowed designers to simulate





**Figure 1.1:** (a) Computer-aided design workflow (b) Building design example in a 3D modeling tool

their building's behavior regarding specific criteria, and, thus get a measurement of its performance [5]. Through this process, called Building Performance Simulation (BPS), designers could easily validate whether their building's performance satisfied the efficiency requirements and, ultimately, optimize their design by iteratively generating multiple variations of the same design, assessing their performance, and selecting the better ones. Albeit being very primitive, architects now had the elementary mechanisms required for optimizing their building's designs.

### 1.1.1 Building Performance Optimization

Building Performance Optimization (BPO), a simulation-based optimization approach, treats the results produced by the simulation tool as the functions to optimize. Although invariably suffering from some degree of imprecision and inaccuracy, using these simulations it becomes possible to estimate the performance of complex designs. Particularly, these estimates are beneficial in designs for which analytical solutions are often very difficult or even impossible to derive [6]. In these cases, the objective function, i.e., the function to optimize, is derived from the simulations' results. These objective functions have a domain which corresponds to the range of acceptable designs as specified by the architect.

A known drawback of simulation-based approaches is the time required to achieve reasonable results for complex systems [7] which is associated with different aspects of the problem, namely (1) its **domain** which, depending on the nature of the problem, might use different methodologies to produce the corresponding estimates (e.g., thermal *versus* structural); (2) its **intrinsic structure** which, depending on the attributes and relations of the system, might lead either to simpler or to more complicated computations (e.g., skyscraper *versus* a small house); and (3) its **analytical model**, which has the essential properties of the system we are trying to simulate and that will be used as input to the simulation tool. Generally, the domain and structure do not change for the same problem, albeit there are numer-

ous ways to produce multiple analytical models. Depending on the level of detail of the analytical model, both the computational time and the result of the simulation might change.

Dar exem-  
plo??

In architecture, the generation of each analytical model is a time-consuming and complicated task. On the one hand, it is often necessary to generate multiple models of the same design because of the simulation tools' specificity, i.e., in order to evaluate a design, each simulation tool requires a specialized model of the same design. On the other hand, simulation tools often yield time-consuming processes, where a single simulation can take up to seconds, minutes, hours, days, or even weeks to complete.

In addition to the simulations' specificity and complexity, architectural designs are inherently complex, thus leading to less predictable objective functions, for which mathematical forms are difficult to formulate [8]. For this reason, information about the derivatives of such functions cannot be extracted and methods depending on function derivatives cannot be used to address architectural optimization problems. As a result, classical gradient-based optimization methods can not be used. Instead, other methods that do not rely on the existence of an explicit mathematical form should be used, i.e., methods that treat the optimization functions as black-boxes, relying uniquely on the outputs of numerical simulations.

### 1.1.2 Algorithmic Design

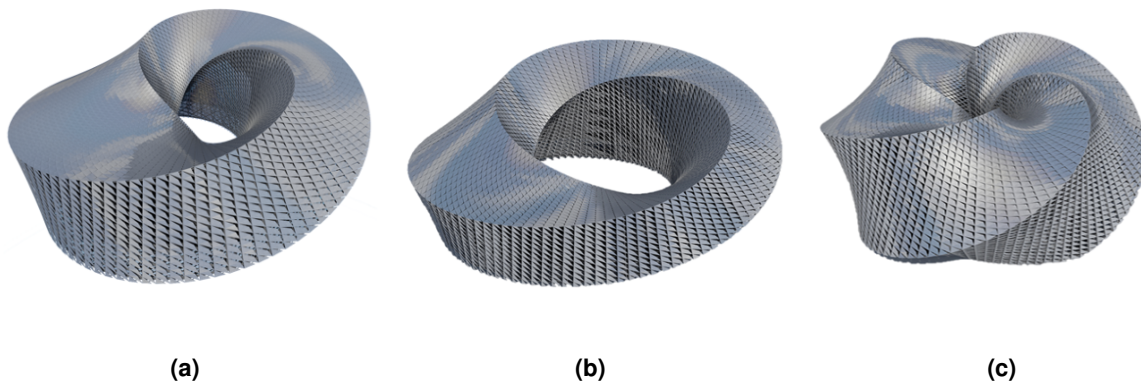
Despite the flexibility provided by CAD and BIM tools, architects often face difficulties when modelling complex geometry. Additionally, when they want to experiment with different design variations, they need to spend a large amount of time to manually make these changes. Since optimization requires evaluating numerous variants of the same design, to manually make these changes would require a lot of effort, hence making the optimization process unviable.



Figure 1.2: Algorithmic-based design workflow

An approach capable of creating forms through algorithms is crucial for overcoming the aforementioned limitations. An example of an algorithm approach is the Algorithmic Design (AD) approach [9], illustrated on Figure 1.2. In this approach, the architect entails an algorithmic description of the design. After executing the algorithm, the corresponding 3D model is created in the CAD or BIM tool. Algorithmic approaches are inherently parametric, thus enabling the generation of different variations of the same design by making simple modifications to the values of the parameters [10]. As an example consider the algorithmic design of the Astana's National Library from BIG architects, illustrated in Figure 1.3(a). Initially, the architect selects an AD tool providing the necessary design primitives. Then, he uses these


primitives to create a computational program enclosing the relative relations among the different design elements, so that when a modification occurs in one element, that modification is propagated throughout that element's dependencies. In the end, the architect creates a procedure responsible for creating the whole design `make-astana(radius, nturns)`, which when executed will produce the corresponding Astana 3D model. Using this procedure, he can easily generate different variations of the Astana by simply setting different values for the parameters: `radius`, defining the width of the whole design, and `nturns`, defining the number of turns in the design. Figure 1.3 illustrates the original design and two design variations are represented: (a) the original design (b) design variation generated by doubling the radius of the original design, and (c) design variations generated by doubling the number of turns.



**Figure 1.3:** Astana's National Library Design variations: (a) Original (b) Larger diameter (c) Two Moebius Turns

Only recently has the algorithmic paradigm begun to settle in the architectural practice. The requirement for programming knowledge is often an obstacle to the adoption of these approaches, since it requires larger initial investments for the architect to learn how to program. Despite the larger initial investments, the benefits obtained with algorithmic approaches surpass the ones obtained when using CAD or BIM tools for the design of complex buildings. Particularly, the initial investments can be quickly recovered when the need for the incorporation of changes arises or when it becomes necessary to experiment different design variations [10]. This is especially important when facing design processes characterized by continuously changing constraints and requirements of the design. In these scenarios, a manual-based approach requires to constantly change the design by hand, thus incurring in a dreadful and tiresome process, whereas an algorithmic approach enables the effortlessly generation of a broader range of design solutions, as well as the easy modification of the algorithmic model to comply with new requirements. As a result, with algorithmic approaches, architects are able to explore larger regions of the design solution space, as well as to explore innovative solutions which were not previously considered due to their time and effort complexity [10].

The appearance of AD was crucial for the automation of optimization processes. On the one hand,




AD enables the automated generation of designs variations by simply changing the parameters. On the other hand, its flexibility enables the creation of extensions like Algorithmic Analysis (AA), that automatically generate the analytical model for the simulation tool, and that subsequently collect the corresponding results - the *update-generate-evaluate* process.

### 1.1.3 Algorithmic Analysis

By enabling a faster and broader exploration of the design space, AD approaches inspired designers to experiment higher levels of complexity, hence leading to building designs which are less predictable at different levels [9], such as thermal, lighting, acoustics, among others. Moreover, recent requirements for efficient and sustainable buildings led to the demand for buildings that not only are well-designed, but also have a good performance at those levels.

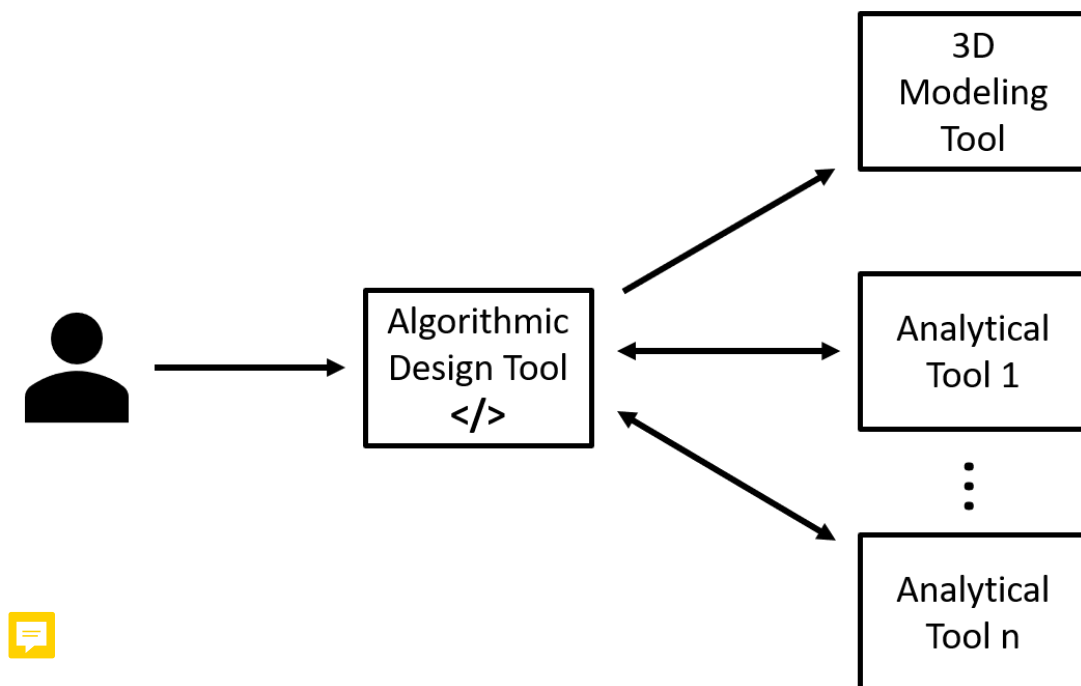
Nowadays, most of the available simulation-based analysis tools are single-domain toolsets, each analysing different parameters within their domain [5], i.e., while a lighting analysis tool measures day-light and glare coefficients, an energy simulation tool measures other coefficients related to thermal, energy consumption, and Heating, Ventilation, and Air Conditioning (HVAC) systems. Unfortunately, this often implies the production of different analytical models for the corresponding simulation tools. Moreover, the 3D models produced by 3D modeling tools are generally dissimilar to the specialized models required by each analytical tool. To evaluate the design performance on different domains (e.g., lighting, energetic, structural), the corresponding analytical models have to be produced either by hand, or through translation processes that convert generic 3D models into specialized models required for analysis.



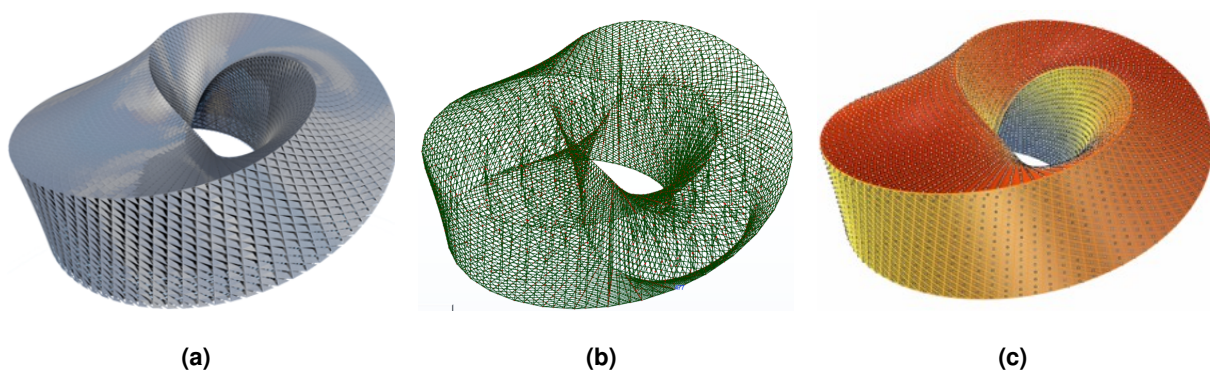
Currently available techniques for the production of analytical models exhibit a few limitations, including: (1) despite the existence of tools that attempt to convert a 3D model into the corresponding analytical model, this conversion is frequently fragile and can cause loss of information or errors; (2) hand-made analytical models might be a more faithful representation of the original model but they require a considerable amount of effort to create; (3) ideally, the results of the analysis would then be used to guide changes in the original design, but these changes require additional time and effort to implement, as does redoing the analysis to confirm the improvements. This explains why performance analyses are typically postponed to later stages of the design process, only to assess the satisfaction of the performance requirements.

To overcome the time and effort limitations associated with the production of analytical models, one can exploit the ideas of algorithmic approaches to automatically generate the necessary analytical models from an algorithmic description. AA is an extension to the AD approach that besides enabling the automatic generation of analytical models from a design's algorithmic model, also automates the setup of the analysis tool and the collection of its results [11]. Following this approach, in an AD tool, the architect

creates the algorithmic model that describes his design's intents and then changes its configurations as to reflect the analysis tool to use, for example, by changing the values of the configuration parameters. Figure 1.4 illustrates the AD and AA design workflow. Note that, even though the abstract description of the design is the same for 3D and analytical models, the produced models can be very different, e.g., a truss is represented as a graph of nodes and edges when submitted for structural analysis, whereas for lighting analysis the truss is represented by its surfaces, and, in the 3D model, the truss is represented by a set of masses representing its bars and nodes. Considering the Astana's National Library design, Figure 1.5 illustrates the distinction among its different models.



**Figure 1.4:** Algorithmic Design and Analysis design workflow



**Figure 1.5:** Astana's National Library design and analytical models: (a) 3D model, (b) Robot's structural analysis model, (c) Radiance's pos-radiation analysis model.

The AA approach is able to enhance performance-based design approaches, as it provides means to effortlessly perform analysis throughout the whole design process, instead of just at final stages. Depending on the performance requirements, architects might need to use different analysis tools: (1) for daylight analysis, DAYSIM and Radiance are very popular among the community, (2) for energy simulations, EnergyPlus, TRNSYS and DOE-2 are widely used [12], (3) for structural analysis, Robot Structural analysis is a well-known reputed tool, and (4) Olive Tree Lab and Pachyderm Acoustical Simulation are examples of good acoustic analysis tools.

Together with AD, AA makes it possible to easily implement automated optimization processes, as it provides the mechanisms to quickly update a design, to generate the corresponding analytical model, and, finally, to automatically evaluate the design in an analytical tool. The optimization processes use the results produced by performance analysis tools for different variations of the design as the functions to optimize.

#### 1.1.4 Architectural Optimization Workflow

In this section, we discuss the different tasks involved in an AD-based design workflow optimization process. We adopt the algorithmic-based workflow due to its growing popularity among the architectural community [?], its parametrics models inherent flexibility, and its ability to automate the *update-generate-evaluate* process.

When following an algorithmic-based design approach, the architect idealizes a design which he ought to produce in the corresponding AD tool. For this purpose, he creates the computational program, defining the parameters that represent the degrees of freedom in his design, i.e., the parameters which he is willing to manipulate in order to achieve more efficient designs. After the conception of the design's algorithmic model and provided the values for the parameters, the AD tool generates 3D or analytical models for visualization and performance analysis purposes, respectively. Optionally, the architect may decide to optimize his design according to some particular aspects, potentially leading to the exploration of design solutions that were not previously considered. In that case, the optimization algorithm explores different design candidates, using the results produced by simulation tools as the functions to optimize. The execution of the optimization algorithm then yields optimal (or near optimal) design solutions.

Considering the previous view of an algorithmic-based design workflow, we identify four key dimensions in an optimization process ~~within that workflow~~:

1. **Analytical models:** when the optimization algorithm specifies a candidate design, i.e., a concrete configuration for the parameters of the model, analytical models are automatically generated by the AD tool. These models are then used as input for the corresponding analytical tools. These models can be improved, either through simplification of the analytical models, or by enriching them with context information. The former enables the simplification of the analysis itself and



potentially reduces the simulation time, by providing an equivalent but simpler model to the tool, whilst the latter enables the attainment of a more detailed and realistic simulation, which is not always possible due to limitations in the AD tool.

2. **Optimization algorithms:** the algorithms that explore the design space in the quest for optimal (or near optimal) solutions. These algorithms use the results obtained from performance analysis of different design variations as the functions to optimize, i.e., as objective functions. Generally, the algorithm uses these inferred functions to guide the search for optimal solutions. The time complexity of the algorithm is typically ~~conditioned on the~~ number of function evaluations. In architectural design, these functions entail time-intensive simulations, thus instilling optimization processes that may take minutes, hours, days, or even weeks to complete.
3. **Intelligibility of Results:** Cichocka et al. identify the demand for intelligibility of the optimization processes within the architectural community [3]. Having access to an explanation regarding the quality of a design solution allows architects to make more informed decisions. With these explanations, the architect can not only provide valuable arguments for its implementation, but also, depending on the quality of the explanations, learn with the process, thus fostering more efficient and faster future designs.
4. **Interactivity and Visualization:** interactive and visual aspects are highly important features in the context of an optimization process [?]. On the one hand, an interactive optimization process enables its user to transfer knowledge about the problem at hand, for instance, by adding or removing constraints or by exploring different, yet unexplored regions of the design space, hence potentially increasing this process' performance. On the other hand, optimization processes providing better visualizations and representations of their own evolution can present their users with better feedback about the course of the search. This feedback is important, as it also allows the comparison of variable-objective correlations and the making of more informed decisions about the optimization process itself, e.g., whether the evaluations made so far suffice or if the algorithm is converging to non-conventional designs that he refuses to accept.

## 1.2 Goals

## 1.3 Organization of the Document

This thesis is organized as follows: Chapter 1 interdu vel, tristique ac, condimentum non, tellus. In chapter 2 curabitur nulla purus, feugiat id, elementum in, lobortis quis, pede. In chapter 3 consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Chapter 4 morbi egestas,

urna non consequat tempus, nunc arcu mollis enim, eu aliquam erat nulla non nibh in ?? . Chapter 5  
suspendisse dolor nisl, ultrices at, eleifend vel, consequat at, dolor.



# 2

## Background

### Contents

---

2.1	Single-Objective Optimization . . . . .	15
2.2	Multi-Objective Optimization . . . . .	15

---



One important classification is regarding the cardinality of the solutions sought by optimization processes, thus yielding the continuous and discrete optimization categories. In the former, the optimal solutions lie in a potentially infinite set of candidate solutions, whereas in the latter, the optimal solutions lie in a finite set. Optimization problems can also be classified as constrained or unconstrained, depending on whether the models explicitly define constraints or not.

Optimization can also be distinguished in terms of the aim of the search that is performed, particularly, whether it is global or local. In local optimization the search process strives to find a solution that is locally optimal, i.e., for which its value is better than all other points in its vicinity. The points that satisfy the previous property are known as local optima. On the other hand, there are optimization processes that strive to find the globally optimal solutions, i.e., the best of all the local optima.

## 2.1 Single-Objective Optimization

Cras dictum. Maecenas ut turpis. In vitae erat ac orci dignissim eleifend. Nunc quis justo. Sed vel ipsum in purus tincidunt pharetra [?]. Sed pulvinar, felis id consectetur malesuada, enim nisl mattis elit, a facilisis tortor nibh quis leo. Sed augue lacus, pretium vitae, molestie eget, rhoncus quis, elit [?]. Donec in augue. Fusce orci wisi, ornare id, mollis vel, lacinia vel, massa. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas..

### 2.1.1 Derivative-Free Optimization

### 2.1.2 Optimization Tools in Architecture

#### 2.1.2.A Galapagos

#### 2.1.2.B Goat

## 2.2 Multi-Objective Optimization

Nunc tincidunt convallis tortor. Duis eros mi, dictum vel, fringilla sit amet, fermentum id, sem. Phasellus nunc enim, faucibus ut, laoreet in, consequat id, metus. Vivamus dignissim [?]. ?? is automatically compressed to fit text width. You can use <https://www.tablesgenerator.com> to produce these tables, and then copy the  $\LaTeX$  code generated to paste in the document.

### **2.2.1 Experimental Approach**

### **2.2.2 Priori Articulation Approach**

### **2.2.3 Pareto-Based Approach**

### **2.2.4 Metrics for Multi-Objective Optimization**

### **2.2.5 Optimization Tools in Architecture**

#### **2.2.5.A Octopus**

#### **2.2.5.B Opossum**

#### **2.2.5.C Optimo**

Cras lobortis tempor velit. Phasellus nec diam ac nisl lacinia tristique. Nullam nec metus id mi dictum dignissim. Nullam quis wisi non sem lobortis condimentum. Phasellus pulvinar, nulla non aliquam eleifend, tortor wisi scelerisque felis, in sollicitudin arcu ante lacinia leo.

# 3

## Solution

### Contents

---

3.1 Architecture Overview . . . . .	19
3.2 Architecture Design Requirements . . . . .	19
3.3 Architecture Design Implementation . . . . .	19

---



Donec gravida posuere arcu. Nulla facilisi. Phasellus imperdiet. Vestibulum at metus. Integer euismod. Nullam placerat rhoncus sapien. Ut euismod. Praesent libero. Morbi pellentesque libero sit amet ante. Maecenas tellus. Maecenas erat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

## **3.1 Architecture Overview**

## **3.2 Architecture Design Requirements**

### **3.2.1 Problem Modelling**

### **3.2.2 Simple Solver**

### **3.2.3 Meta Solver**

## **3.3 Architecture Design Implementation**

### **3.3.1 Problem Modelling**

### **3.3.2 Simple Solver**

### **3.3.3 Meta Solver**





# 4

## Evaluation

### Contents

---

4.1 Qualitative Evaluation . . . . .	23
4.2 Quantitative of Applications . . . . .	23

---



- Relembrar o objectivo do trabalho e dizer como o vamos avaliar de um modo geral introduzindo os proximos subcapitulos.

## **4.1 Qualitative Evaluation**

- Number and Heterogeneity of Available algorithms - Differences / Benefits / Disadvantages when compared to Grasshopper's frameworks

## **4.2 Quantitative of Applications**

- Dizer que de um modo geral começámos de forma incremental por considerar problemas single-objective, nomeadamente a casa da ericeira, que remonta a primeira publicação. Depois evoluimos para a avaliação bi-objetivo de dois casos de estudo reais - Pavilhão Preto para exposições e de uma arc-shaped space frame.

- Comentar a facilidade c/ que alguém que já tem um programa AD consegue acoplar optimização a AD.

### **4.2.1 Ericeira House: Solarium**

### **4.2.2 Black Pavilion: Arts Exhibit**

#### **4.2.2.A Skylights Optimization**

#### **4.2.2.B Arc-shaped Space Frame Optimization**



# 5

## Conclusion

### Contents

---

5.1 Conclusions . . . . .	27
5.2 System Limitations and Future Work . . . . .	27

---



Pellentesque vel dui sed orci faucibus iaculis. Suspendisse dictum magna id purus tincidunt rutrum. Nulla congue. Vivamus sit amet lorem posuere dui vulputate ornare. Phasellus mattis sollicitudin ligula. Duis dignissim felis et urna. Integer adipiscing congue metus.

Rui Cruz  
You should  
always  
start a  
Chapter  
with an in-  
troductory  
text

## 5.1 Conclusions

## 5.2 System Limitations and Future Work

### 5.2.1 Optimization Algorithms

### 5.2.2 ML models

### 5.2.3 Constrained Optimization

Aliquam aliquet, est a ullamcorper condimentum, tellus nulla fringilla elit, a iaculis nulla turpis sed wisi. Fusce volutpat. Etiam sodales ante id nunc. Proin ornare dignissim lacus. Nunc porttitor nunc a sem. Sed sollicitudin velit eu magna. Aliquam erat volutpat. Vivamus ornare est non wisi. Proin vel quam. Vivamus egestas. Nunc tempor diam vehicula mauris. Nullam sapien eros, facilisis vel, eleifend non, auctor dapibus, pede.





# Bibliography

- [1] M. W. Online. (2018) Merriam Webster Online - Optimization Definition. [Online]. Available: <https://www.merriam-webster.com/dictionary/optimization>
- [2] J. Nocedal and S. J. Wright, *Numerical optimization*, 2011, no. 2.
- [3] J. M. Cichocka, W. N. Browne, and E. Rodriguez.
- [4] B. Ferreira and L. António, "Generative Design for Building Information Modeling," *Proceedings of the 33rd Education and research in Computer Aided Architectural Design in Europe Conference*, vol. 1, pp. 635–644, 2015.
- [5] A. Malkawi and B. Kolarevic, *Performative Architecture: Beyond Instrumentality*, 2005, vol. 60, no. 1.
- [6] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by Direct Search - New Perspectives on Some Classical and Modern Methods," *Society for Industrial and Applied Mathematics*, vol. 45, no. 3, pp. 385–482, 2003.
- [7] A. M. Law and W. D. Kelton, *Simulation modeling and analysis*, 1991, vol. 2.
- [8] V. Machairas, A. Tsangrassoulis, and K. Axarli, "Algorithms for optimization of building design: A review," *Renewable and Sustainable Energy Reviews*, vol. 31, no. 1364, pp. 101–112, 2014.
- [9] R. C. Branco and A. M. Leitão, "Integrated Algorithmic Design: A single-script approach for multiple design tasks," *Proceedings of the 35th Education and research in Computer Aided Architectural Design in Europe Conference (eCAADe)*, vol. 1, pp. 729–738, 2017.
- [10] A. Leitão, L. Santos, and R. Fernandes, "Pushing the Envelope: Stretching the Limits of Generative Design," *Blucher Design Proceedings*, vol. 1, no. 7, pp. 235–238, 2014.
- [11] R. Aguiar, C. Cardoso, and A. Leitão, "Algorithmic Design and Analysis Fusing Disciplines," pp. 28–37, 2017.
- [12] A.-T. Nguyen, S. Reiter, and P. Rigo, "A review on simulation-based optimization methods applied to building performance analysis," *Applied Energy*, vol. 113, pp. 1043–1058, 2014.

