Digital Design Optimisation: New Methods and Tools for Design and Manufacturing of Architectural Objects

Dirk Gerard Celina Robert Loyens

UMinho | 2012

Universidade do Minho
Escola de Arquitectura

Dirk Gerard Celina Robert Loyens

Digital Design Optimisation:
New Methods and Tools for Design and Manufacturing of Architectural Objects

Junho de 2012

Universidade do Minho
Escola de Arquitectura

Dirk Gerard Celina Robert Loyens

Digital Design Optimisation:
New Methods and Tools for Design and Manufacturing of Architectural Objects

Tese de Doutoramento
Arquitetura e Tecnologia

Trabalho efectuado sob a orientação do
Professor Doutor Paulo Jorge Figueira de Almeida Urbano de Mendonça
Professor Doutor Ferrie Wander Joseph van Hattum

Junho de 2012

For Maria José, Dinis and Matilde.

Acknowledgements

I gratefully acknowledge the valuable support and feedback I have received from my supervisors:

Professor Doutor Paulo Jorge Figueira de Almeida Urbano de Mendonça
Escola da Arquitetura – Universidade do Minho

and

Professor Doutor Ferrie Wander Joseph van Hattum
Escola da Arquitetura – Universidade do Minho

For the indispensable and generous collaboration throughout this research, I would like to give my special and respectful thanks to Professor Doutor António Gaspar Lopes da Cunha from the Departamento de Polímeros – Universidade do Minho.

With thanks to Inês Melo and Sérgio Afonso (ESAD) for collaboration on the graphical layout of this thesis and the creation of some very nice figures, and to Ann for helping me out with the programming of the GUI.

The author acknowledges the financial support received from the Portuguese Science Foundation (**FCT**) under grant: **SFRH/BD/44600/2008**., co-financed by the European Social Fund (**FSE**) under the program: Programa Operacional POTENCIAL HUMANO (**POPH**) within the Quadro de Referência Estratégico Nacional (**QREN**).

FCT Fundação para a Ciência e a Tecnologia MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

UNIÃO EUROPEIA Fundo Social Europeu

POPH PROGRAMA OPERACIONAL POTENCIAL HUMANO

QREN QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL PORTUGAL 2007.2013

Digital Design Optimisation: New Methods and Tools for Design and Manufacturing of Architectural Objects.

Abstract

The integration in the design process of high-level objectives, such as those requirements directly related to functional performance and lightness or the efficient use of materials, is especially difficult and can no longer be directly and intuitively perceived by the designer. This is particularly true in the conceptual phase of a design process. Methods and design tools that take in account personal preference and cultural acceptance, and can combine the interactive behavior inherent to conceptual designing with the formal rigor of optimisation are therefore desirable.

The aim of the work presented in this dissertation is to develop and test a method that could be employed to model and manage the design process individually, adapted to a particular design problem and along with the personal preferences of the designer. The method is part of a digital design process where simulation and analysis does not only support the project development process, but interacts and contributes with novel proposals, promotes a more creative exploration of the solution space and aims to integrate computational models in the reasoning process and the activities of the designer during the complete design process

A digital design process allows for integration of simulation and analysis software right in the beginning of the project, but it will be the task of the designer to build his own software construct combining all necessary software programs with the intent to introduce optimisation efficiently and goal orientated into the design process. In this thesis a design method is presented based on a software construct combines parametric design with evolutionary principles with the intent to maximise explorative search in an iterative design process. This application consists of a loose combination of commercially available software programs and property scripts united towards a common goal.

The ability of this method to interactively assist the designer during the design process is demonstrated and applied to the conceptual design of several case studies, in the form of shading devices. It is concluded that optimisation can be introduced at the very beginning of the process of designing and optimisation reveals to be helpful and increasingly needed for an effective design process when constraints and boundary conditions cannot be easily evaluated by a conventional intuitive process and general domain knowledge. This advance of performative orientated design can only be viable within a within a digital supported methodological approach.

Design Digital e Otimização: Novos Métodos e Ferramentas para o Design e Produção de Objetos Arquitetónicos.

Resumo:

A integração no processo de design de objetivos meta projeto, tais como os requerimentos relacionados com a leveza, o desempenho funcional, ou o uso eficiente de materiais, são especialmente difíceis de compreender e já não podem ser entendidos direta e intuitivamente pelo próprio designer. Isto revela-se particularmente na fase conceptual do processo de desenvolvimento. Métodos e ferramentas de design que conseguem integrar a preferência pessoal do designer, fatores culturais e que podem ao mesmo tempo combinar um comportamento interativo - inerente ao design conceptual - com o rigor formal da otimização são, portanto, desejáveis.

O objetivo do trabalho apresentado nesta dissertação é desenvolver e testar um método que pode ser modelado e organizado individualmente conforme o próprio processo de desenvolvimento de um designer. Este método pode adaptar-se a qualquer problema de design em particular e pode ser totalmente construído conforme as preferências pessoais e as necessidades do designer. O método é parte de um processo de design digital, onde a análise e a simulação não apenas apoiam o processo de desenvolvimento do projeto, mas também interagem como processo de exploração e contribuem com propostas para soluções diferentes. Pode desta forma contribuir para uma exploração mais completa e mais criativa do espaço de soluções. Esta abordagem quase completamente digital promove, ao mesmo tempo, a integração de modelos computacionais no raciocínio e nas atividades do designer durante o processo de design completo.

Um processo de design digital permite a integração de *software* de simulação e de análise logo no início do projeto. No entanto, deverá ser a tarefa do designer construir o seu próprio *software*, combinando os programas e os *scripts* necessários com a intenção de introduzir a otimização no seu processo de design de uma maneira eficaz e eficiente e, claro, em conformidade com o objetivo específico do processo. Nesta

tese, é apresentado um método baseado na combinação de *softwares*, que juntamente com uma geometria paramétrica e com princípios evolutivos, permitam maximizar a pesquisa exploratória do espaço das soluções num processo de design iterativo. Este método consiste numa combinação, com um objectivo comum, de programas de *software* disponíveis no mercado e de *scripts* desenvolvidos pelo próprio designer.

A capacidade deste método apoiar o designer, de uma forma interativa, durante a fase conceptual do processo de design é demonstrada e aplicada a um projeto conceptual de sombreadores. Concluiu-se que a otimização pode ser introduzida durante a fase conceptual do processo de design e que a otimização se revela não só útil, mas cada vez mais necessária para um desenvolvimento eficaz, quando os constrangimentos e os limites não podem ser facilmente avaliados por um processo intuitivo e convencional baseado no conhecimento geral do domínio. O avanço do design orientado para a *performance* só é possível dentro de uma abordagem metodológica digital.

Table of Contents

List of figures

List of Tables

1 Introduction

1.1 Interactive optimisation

Design and design theory are changing very fast and have changed significantly over the last 20 years. The traditional design approaches based on intuition, visualisation and emotion alone are unable to fully provide the designer with the understanding of how the final solution will behave. When the designer "has to make all kinds of decisions without adequate information and knowledge, his task is experienced as depressing" (Zeiler et al., 2007, p. 4). A good design process must combine the scientific requirements and the aesthetic concerns on an equal basis.

Research in optimisation for design has traditionally developed along two different directions (Gero & Kelly, 2008). On one side, and ever since the beginning of the use of computers in the discipline of design, research has focussed on pure automated design through the application of knowledge-based techniques and the application of artificial intelligence techniques. The practical application of the results of this research however has been limited to a narrow group of very specific design problems, mostly in the field of engineering design, but has not been very successfully applied in a more conceptual design process. On the other hand, results of recent research in other fields has boosted renewed interest and focuses on creating a general methods for design tools (Sarkar, Dong & Gero, 2009), or an approach that combines the interactive behaviour of conceptual designing, the necessary rigor applied through optimisation and a process of learning and reusing information (Caldas, 2005)(Sarkar, Gero & Saunders, 2007)(Turin, von Bulow & Stoufs, 2011)(Bintrup, et al., 2008).

The integration in the design process of high-level objectives, such as those requirements directly related to functional performance and lightness or the efficient use of materials, is especially difficult and can no longer be directly and intuitively perceived by the designer. Consequently they become therefore typically compromised and disconnected from the design process, particularly whenever the conditions under which the object will perform can only be assessed based on expectations, assumptions and imagination (Horváth, 2005). This is particularly true in the conceptual design stage. Unfortunately at this point of the design process, it is experience and intuition which are usually the only guides for evaluation. Only later, in the detailing stages of the design process do designers apply analysing and simulation software (Schwede, 2006b). But design decisions taken in the early stages of the design process determine and impose limits on the potential performance of design objects later in the physical world, much more than those which are taken at later phases of the design process. Therefore the strategies which are followed in the beginning of a design project and the decisions made during those early stages are the most important, although the early decisions are traditionally based on less knowledge about the goals to be attained (Derelöv, 2009). It is quite possible that the conceptual phase is the most significant part of the design process.

It is also in the early stages that the designer has the greatest freedom to explore the overall solution space (Ullman, 2009). However this freedom cannot be fully explored if the designer does not know which lines of thought are worthwhile to pursue, and if the efforts should not be better directed to exploring different concepts (Matthews, 2002).

Designers and architects are most of the time searching and looking for a "satisfactory solution" to their design problems (Simon, 1996, p. 119). Unfortunately the final proposal is almost never the best possible solution and might not even be in terms with the initial design goals and constraints of the project. The designer usually stops exploring as soon as a solution is judged as good enough. Often the reason behind this strategy of "good enough" instead of "perfect" is the lack of better tools for simulation and analyses. Mostly it is just lack of time and insight.

Currently, powerful analysing software has become readily available and allows for computational simulation of the design objects' performance before it is built and available for investigation in the physical world. But applying those specialised software tools requires deep understanding of the design problem and the complex relationships between constraints, requirements and goals within the specific domain. The flawless integration of those tools asks for a new kind of designer, a designer who is computer literate and who can built his own custom software constructs and develop strategies for its use. At the same time the designer has to abandon the traditional paper based design process and start working in a fully digital environment. The convergence between geometry and performance can only be designed, analysed and evaluated with the help of digital technology (Sousa & Duarte, 2005). And it is the full integration of sophisticated and interactive digital media throughout the complete design process that has provoked the emerging of new paradigmatic models of design thinking. Those new theoretical models could be the fundamentals of a new kind of design, digital design where digital techniques couple the principles of performance with the principles of generation (Oxman, 2006).

However relevant, performance issues differ from one design project to another and testing in the physical world or under controlled laboratory conditions cannot or should not be ruled out of the design process. If the construction of physical prototypes was a cumbersome and labour intensive endeavour in the past, contemporary CNC-technology enables the designer to built his own models and prototypes effortless and almost on the fly. Furthermore, current rapid prototyping methods and techniques allow for a seamless integration in a digital design process and thus enables for a direct correlation between what can be designed and what can be built (Kolarevic 2005). This allows for a more fundamental awareness of the digital tools available to the designer, besides the traditional goal of representation for visualisation.

1.2 Aims and scope of research

Designers and architects don't like to be told how to work and can be very aversive towards any collaboration between the human designer and computational constructs. The obligation to correspond to strict constraints is experienced as a limitation on the quality outcome of their creative process. Those limits are also underlying the objectives of software tools which have been developed to assist the designer in the design process: almost all are targeted to the technical or the knowledge base part of the design process. Very few of those tools are sensitive to the broadminded creative goals of the design process.

Computational simulation is mainly used to determine the performance of completed designs after their development. In the early phases of the traditional or conventional design process designers use and apply heuristics learnt through personal experience. They transfer design knowledge from past experiences into current ones (Sarkar, Gero & Saunders, 2007), but have no support or assistance during the conceptual phase of the design process although design decisions made during the early stages have a great impact on the quality of the final solution (Derelöv, 2009).

The aim of the work presented in this dissertation is to develop and test a method that could be employed to model and manage the design process individually, adapted to a particular design problem and along with the personal preferences of the designer. The method will be part of a digital design process where simulation and analysis does not only support the project development process, but interacts and contributes with novel proposals, promotes a more creative exploration of the solution space and aims to integrate computational models in the reasoning process and the activities of the designer during the complete design process.

The intention is to develop a software construct by integrating optimisation algorithms, simulation software and rapid prototyping techniques in the very beginning of the design process of architectural objects. The focus is on real applications where virtual prototyping and simulation software is used to analyse the properties of a system during the design process, and where physical prototypes can give real world feedback and increased insight in the design problem.

This thesis takes a research-through-design approach to investigate how digital design thinking can mould and shape a digital design process. A framework will be developed and tested so that optimisation tools can be constructed by the proper designer not only as a design strategy, but above all as a design method. It will also examine how, within a digital design environment, the possibilities and the role of optimisation techniques can provide valuable information to be used in the early stages of the design process. By doing so, this thesis explores new relationships between the designer, information, the process and the object, and thus explores the possible distinctive character of digital design thinking (Oxman, 2006).

This research is limited to the study of simple architectural components, which can be developed in existing CAD and CAAD environments, and will also be limited to the integration of those requirements that can be simulated by existing software. This research project does not aim to be universally applicable and, when material characteristics are required, will be limited to the use of polymer composites, as this group of materials offers the widest variety of technical characteristics and geometrical freedom, thus momentarily not considering other more conventional materials such as wood, glass or steel. However, the results of the research are applicable with slight adaptations to these materials as well.

1.3 Structure of this thesis

This thesis is organised in 5 chapters. The first chapter gives an introduction to the work, outlines its context and illustrates some of the key concepts used and elaborated further on in the other chapters.

Chapter 2 shortly reviews the proceedings and historical paradigms in design. Those paradigms are a starting point for the review of the historical development of optimisation in design and in architectural design, and are a key element in the description of a new field recognised as digital design. Some conclusions will be drawn about the special role that the field of digital design will occupy in the future development of the praxis of design. The second part of this chapter is concerned with several methods, approaches and algorithms, which have been researched and proposed in different design fields such as mechanical design and architectural design.

In Chapter 3 the basic principles are identified and the general concept of a descriptive model for interactive optimisation is described. Each of the desirable and necessary requirements of the design method proposed in this study is formulated, described and discussed. Different components of the early stages of the design process are described and the relationship with optimisation discussed. This chapter also includes a short reflective analysis about design and interactive optimisation. The different practical components of the software construct are presented and explained. Each of the parts of the software construct is briefly demonstrated and tested. This chapter also serves as an introduction to the case studies presented in chapter 4.

In Chapter 4 the proposed design method is assessed and tested. Two case studies were built and executed. The results from those case studies are illustrated in detail and the results briefly analysed and commented.

The main objective of the first case study is the construction of a fully functional software construct which can simulate the working of the proposed method and its possible integration in a design process, and with the focus on integration of the CAD software and the simulation and analyses software. A second objective will explore the possibilities of a seamless transfer of information between the different parts of the software construct. This first case study is divided in two parts. In the first part the integration of all the scripts and all the software will be tested and explored with different optimisation strategies and with the optimisation of two straightforward functional parameters – daylighting and surface area of a *structure* – together with the input of the aesthetical preferences of the designer. In the second part the software construct will be briefly tested towards the same optimisation criteria but also including structural optimisation.

The main objective of the second case study is the construction of a fully functional software construct for interactive optimisation which will be integrated, as a method, in a complete real world design process. The fully digital developed design object will subsequently be constructed in the physical real world for testing and validation of the digital results

Chapter 5 concludes the study, the results from the different case studies are compared and the proposed concepts are discussed and evaluated. The assessment and practical application of the proposed design method will be discussed in detail. Some suggestions are also provided for future research work.

1.4 Terms used in this thesis

In this paragraph some of the key concepts and terms used throughout this thesis will be briefly explained and discussed. It is not meant to be a glossary but a useful reference about how different concepts are related.

This thesis makes the assumption that designing is a sequence of acts which can be described through procedures, this way it will become a method and not a tool. The software construct demonstrated in this thesis was elaborated with the same guiding principal.

Soft computing is hard to define but "can be seen as a series of techniques and methods so that real practical situations could be dealt with in the same way as humans deal with them, i.e. on the basis of intelligence, common sense, consideration of analogies, approaches, etc." (Verdegay, Yager & Bonissone, p. 848). It finds its origins in the necessity to explore the tolerance for imprecision, uncertainty, and partial truth (Zadeh, n.d.). The methods and techniques of soft computing have entered the research agendas of many other disciplines, such as biomedicine, economics, logistics, etc., where finding or providing exact solutions to complex problems is impossible or extremely difficult. These are also characteristics which one can apply to the way proposals and design candidates are developed in the early stage of a design process.

Evolutionary algorithms (EA) encompasses evolution strategies, evolutionary programming, genetic programming and genetic algorithms, and are by far the most popular and most applied technique used in research in optimisation, also in the field of architecture. Among other advantages, EA are cheap and easy to implement, they are robust and can thus be applied to all kinds of problems and one does not need a deep mathematical understanding of the problems to which they are applied (Fontes & Gaspar-Cunha, 2010).

The first part of a design process in usually called the "front end". Since this is the phase where ideas and generic proposals proliferate and everything is still uncertain and not clear at all, some authors talk about the Fuzzy Front End where logic and reason do not yet eliminate creative exploration of ideas. The method presented in this research can therefore refer to a Fuzzy Front End Optimisation (FFO), and will describe the entire set of actions and decisions of a designer or an architect specific to the early stage of a design process where concepts are explored through iteration.

When using EA for multi-objective optimisation, niching and other techniques are applied to a specific EA with the goal of streamlining the process and maintaining diversity of certain properties within the population with the objective to converge into multiple good solutions.

The terms digital and analogue used in this thesis distinguish between computer-supported (digital) processes and non-computer-supported (analogue) processes, also recognised as paper-based processes. Besides this distinction, directly related to the pragmatic environment of the design process, digital and analogue can refer also to Otl Aicher´s view on design and his philosophy of making: the basic thought that making and thinking are intrinsically interdependent of each other. In his approach he argues that the *concrete and analogues* comes before the *abstract and digital* (Aicher, 1994).

In this thesis "intuitive" is used and understood as knowing or perceived by intuition, meaning attaining to direct knowledge or cognition without evident rational thought and inference (Merriam-Webster, 2003).

2 Digital Design and Optimisation

In this chapter the main historical methodological paradigms in the field of design are briefly reviewed and discussed. This overview of design paradigms is necessary to contextualise and to explain key concepts of the design model presented later in this thesis. The role of optimisation in design is also approached and briefly reviewed. This review serves to demonstrate different viewpoints and some key positions about optimisation in design. All those reviews are necessary to recognise a new field in design with a specific and distinctive character: digital design and digital design thinking.

The second part consists of a short review of some practical research in optimisation in the field of design. Some of the work of other researchers will be briefly described, and some of the different methods and approaches - although still scarcely researched, proposed and applied in the field of product design and architectural design, but more frequently in the field of engineering design - will be described.

2.1 Digital design

The use of computers in the design process started more than 4 decades ago and has been widely researched and discussed meanwhile. The first use of a computer to generate an architectural representation for appraisal was in 1966 (Kolarevic, 2004). One of the first tools for digital performance analysis was developed in 1973 (Maver, 2000). Further development of computer technology, computing power and the emerging of a new kind of architecture evolved in two different, but not always very distinguished approaches on understanding the role of computers in the design process.

A first approach reduces the use of computers to a simple design tool with the main intention to speed up the design and development process and in some way substitute the human designer in those tasks which are repetitive and cumbersome. Drafting and modelling software are examples of the use of computational power replacing traditional means (pencil and paper) but without fundamentally changing the task of drafting or modelling. It is still the designer who has to give detailed instructions (Kalay, 2006). In the same spirit the computer has also been used to assist the designer in predetermined tasks such as calculating complex geometrical operations, such as described and illustrated by the work of design engineers at Bollinger + Grohmann, Dominique Perrault (Fig. 1) and Coop Himmelb(l)au (Menges, 2006)(Bollinger, Grohmann & Tessmann (2008) or descriptions of the use of computers in the building of the Sydney Opera House (Neil, 2006).

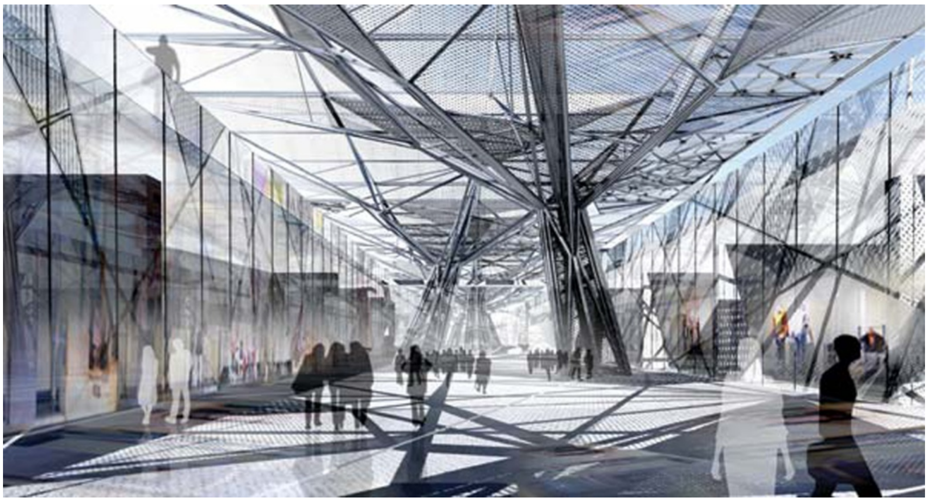


Fig. 1 – Underground roof Piazza Garibaldi Naples 2007 (Dominique Perrault).

Another approach is the use of computers as a medium. This implies that the computer assists the designer in the creative process, providing him with a new understanding of the design problem by presenting unexpected solutions, such as illustrated by the use of morphogenetic design (Hensel, Menges, & Weinstock, 2006). It is this approach, which is of more interest for designers, and it is this search for the role of computer technology in design, which will lead to other and different solutions in contemporary design and architectural design.

2.1.1 Methodological Design Paradigms

Design is, comparatively to other disciplines, a rather new field of research and only during the last century more general frameworks for design and design theory have been researched, developed and described. In the very beginning of design research, design was essentially modelled as a basic problem solving process, similar to the way people solve simple everyday problems (Burdek, 1991). One of the earliest programmatic models of the design process, in the beginning of what later would be called the "Design Methods Movement", was described by Asimov (Bayazit, 2004) who divided all design processes in three basic stages: analysis – synthesis – evaluation, eventually further elaborated and graphically depicted through nested loops. This basic stage, rather than providing a model of the design process, describes a basic iteration cycle, a process which occurs with different intensities and with different frequencies all over and during a design process.

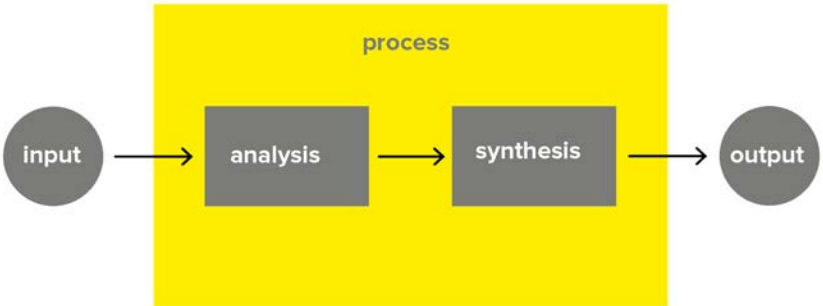


Fig. 2 – Design Process Archetype according to Koberg and Bagnall (Dubberly,).

Thus, at first, design was recognised as nothing more than a simple linear process, such as depicted in figure 2 (Dubberly, 2005), and, only later in the twentieth century, the theories of problem solving were introduced to the design discipline. It was this understanding that design is "problem solving" that led to the phase models of the design process and to elaborate descriptive and prescriptive models (Jones, 1992)(Roozenburg & Eekels, 1995), and it was Herbert Simon who provided design with a supportive framework for the paradigm of technical rationality (Simon, 1996). Engineering design, less open ended and usually with a clear and explicit objective, has profited a lot from this research about design and development methods (Ulrich & Eppinger, 2007) and research in design methods continues till today (Donaldson, 2006).

But the results from cognitive studies of designers during design activities indicated that design does not only involve a search process. The designer is constantly confronted with situations of uncertainty and instability which are very different and most of the time conflicting. Clearly, design is not a typical process of problem solving of well formulated problems where one can use a rational, systematic and scientific approach. Thus, problem solving for a designer does also imply constant adapting or searching for a solution space, while at the same moment reformulating the problem statement. These observations evolved into a different design paradigm which has been coined as design-by-learning. Designers think about what they are doing while they are doing it. Design is now recognised as a process of "reflection-in-action" (Schon, 1991) and described as an iterative process of experimentation with the goal of producing better understanding of the design problem through the interaction of making and seeing. This is effectively much more a description of how the design process actually occurs in professional design practice. Design should therefore be better approached as an open-ended process, oriented to innovative solutions. Exploration and experimentation are the tools used in the objective for probing for possible solutions. Every possible solution is the basis for an evaluation against any previous design concept and the start of a process of generating new design concepts (Schon & Wiggins, 1992). According to those researchers a designer generates and explores concepts by moving through the solution space.

Design by learning has been further elaborated using an approach to reflection that is more process centred (Gero & Kannengiesser, 2009), more focussed and characterised in quantitative terms, and also suitable for non-routine design problems. This new framework is based on the theory of situatedness and situated cognition originated by Dewey (1896)(Clancey, 1997). Cognition refers to the processing of information and the application of knowledge. The theory of situated cognition argues that knowing is inseparable from doing and that cognition cannot be isolated from its context

(Gallagher, 2008). It states that knowledge is dynamic and a result from subjective interaction. It claims that every human thought and action is adapted to the environment because of what people perceive, how people evaluate their actions and what people physically do, develop together (Clancey, 1997)(Gero & Kannengiesser, 2009). The level of knowledge about the problem changes during the process, learning occurs because new knowledge is gained and old knowledge is restructured. Other researchers in the field of design creativity arrive at similar conclusions and use radical constructivist theory to describe the concept of *Perception in Action* as a methodological design process where the designer challenges stereotypical thinking and consciously searches for the new and the different in the solution space (Tschimmel, 2010). From this point of view Design is no longer 'Problem Solving' but rather a process of 'Decision Making' that implies making choices and framing parameters.

Research in design and artificial intelligence has provided deeper insight in how the design process evolves and what is the role of the designer in that process (Gero, 2006). Humans construct relationships between function (what does), behaviour (how does) and structure (what is). These are based on the experience of the designer and on the interactions with the design object. The understanding of the importance of these relationships for the design process formed the basic structure of a novel model for designing. This model was first developed by Gero (1998) and later further elaborated and refined towards a fully functional framework for design optimisation (Gero & Kannengiesser, 2006). This understanding of the importance and the consequences of interactions are very important for the introduction of optimisation in the conceptual phase of the design process because it can provide an understanding of the necessity of structured iteration loops as a supporting base for interpretation in an optimisation model (Gero & Kelly, 2008) and how problem formulation evolves from a object oriented view to a process oriented view.

Some authors however do criticise the treatment of design as reflective conversation and state that this approach lacks the clarity and the rigour achieved by a more rational problem solving model of design (Dorst &

Dijkhuis, 1995). A design process without a well-defined problem and without a carefully planned design strategy can never be efficient and completely satisfactory. And design can never be pure problem solving alone, but parts of the design process can benefit from a model that structures design on a more rational base. In Dorst's framework, designing is not in itself a subset of problem solving, but problem solving is a subset of designing (Dorst, 2006).

2.1.2 Design and simulation

Computer processing power capacities have improved significantly in the last decade and have become at the same time affordable for any individual designer. A lot of different software applications which can assist the designer in almost every task are available of-the-shelf, and one does not need to be an expert in computer languages and programming to engage in scripting one's own routines or constructing form generators with graphical algorithm editors such as (amongst others) Maya, Grasshopper or Processing. These factors allowed for the complete transition of paper based designing to almost fully digital designing. Thus by developing a project, applying an essentially digital design process based on digital design techniques and implementations, the designer gains the freedom and the possibility to work in interdisciplinary teams using and applying novel universal and integrated technologies (Augenbroe, 2004).

The possibilities which the detailing and the digital augmentation of design representations offer, together with programmed analysis and simulation capabilities of specialised software applications – such as, for example Ecotect 2011 (Fig. 3) used in this study - can now support the design process by offering a better understanding of the behaviour of the designed object in the future physical world. It can be assumed that this would result in a shift of focus guiding the design process. The design process moves away from what was until now essentially *structure centred* to what will now be *performance centred* (Schwede, 2006a). The process which allows this shift in focus is simulation. Simulation is the imitative experimental modelling or representation of the functioning of a system or process by means of the functioning in another, mostly computer based system, in order to get a deeper insight in this functioning (Merriam-Webster, 2003). Strictly speaking, it does not have to be a computer-based method; conventional real world experiments are also widely used and can also supply valuable results and information, as well as validation for supporting simulation results. Schwede (2006a) refers further to research which claims that computational modelling allows for quicker and more detailed simulation and within a fully digital design process, the evaluation of design concepts can be evaluated in a safer way, less expensive and on a better user-interactive manner than similar testing in the real world. Computational simulation can also reduce or even overcome limiting constraints of testing in the physical world and can investigate systems which are too complex to be understood by simple analytical reasoning. It can also describe a system's behaviour and show its spatial properties. In this way it can be used as representation and as a tool for communication, which is very important since most of the time the amount and the structure of the data is much too rich or not suitable for simple verbal communication.

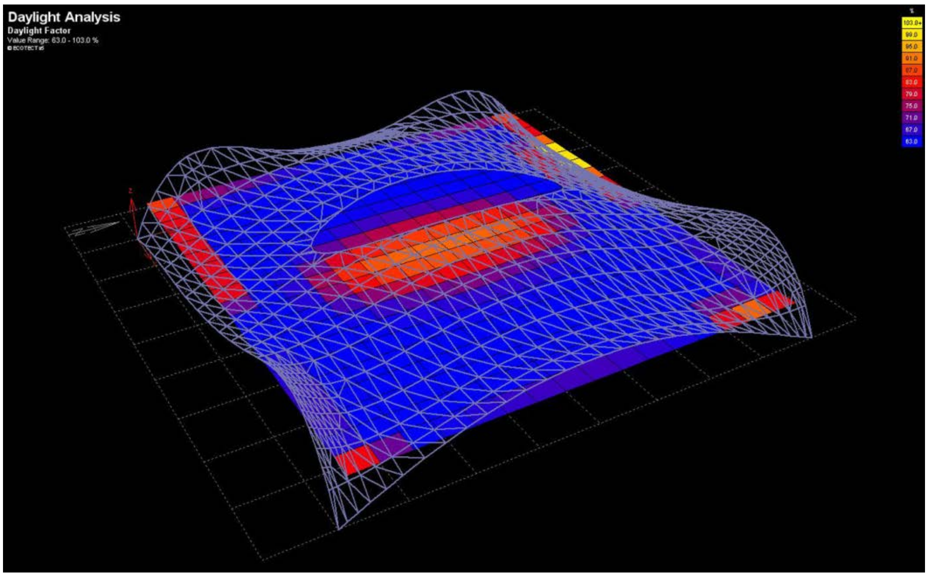


Fig. 3 – Example of tested Daylight Analysis in AutoDesk Ecotect 2011.

Simulation also enables researchers and designers to preview and evaluate systematically how effectively a proposed solution corresponds to prevailing boundary conditions and requirements. Sometimes simulation is applied, before the detailing phase, but most of the time simulation provides information and confirmation during the detailed development of the project.

A new approach on computational simulation has emerged with the development of augmented virtual environments which combine virtual simulation with physical simulation elements. This kind of setup is not only used for training purposes and skill refining, such as those simulation environments used by fire fighters, military or sportsmen. It can also be very useful for scientist, engineers and architects who can enter a virtual space and evaluate its physical structure without actually building that structure (Malkawi, 2004). It can be predicted that in the future this kind of 3 dimensional simulation constructs would comprise entire person-environment-equipment systems with the double function of detailed medium for research and for effective communication between designers, engineers and other professionals (Augenbroe, 2004). Haptic interfaces and so called immersive interfaces (CAVE - Cave Automatic Virtual Environment) systems, originally developed at the University of Illinois (Cruz-Neira, Sandin & DeFanti 1993) are at present used for research in Universities all over the world (Ye & Campbell, 2006) (Bordegoni, Colombo & Formentini, 2006) or (Dunston, et al., 2011), and allow for the development of different approaches to the design process.

2.2 Performance Based Design

Performance based design is an approach in which certain qualitative and measurable objectives are the guiding principles of the design process. In architectural design this may be defined as the exploitation of building performance simulation for the modification of geometrical form towards a predefined objective (Oxman, 2006). Performative design is thus an alternative approach for designing where form, material, structure and performance are understood as inherently related and an integral part of the design process. It should therefore not be mistaken as a simply pragmatic method for solving basic practical problems. Rather, performative design is based on the integration of intangible cultural aspects on one hand and qualitative and quantitative aspects on the other hand (Kolarevic & Malkawi, 2004).

Performance has long been recognised as an important factor in architectural design. Historically it draws on the notions of determinism and functionalism (Kolarevic & Malkawi, 2004). In architecture this can be traced back to Vitruvius and the three classic goals of architecture (Maciel, 2006), and in the past examples of great sophistication and beauty can be found where local resources were applied to provide comfortable conditions for human habitation (Hensel, 2008).

It is however only in the last few decades that performance of design got more attention and started to play a central role in the design process. Research in performative design thinking was pioneered in the 1970's by Tom Maver and his team at the University of Strathclyde in Glasgow with the introduction of the performance analysis tool PACE (Package for Architectural Computer Evaluation - 1973), which allowed for a kind of man-machine interaction converging to a better design solution (Maver, 2000) (Kolarevic, 2004). This research resulted in an important first practical application with the objective of integration of modelling and predicting the performance of buildings at the early stages of decision-making in the design process.

Further research in this field of performance-oriented design is at the basis of a fully developed paradigm that combines different holistic and integrated processes and aspires to be the design solution for an alternative model for sustainable development. Within the paradigm of digital design, performative design represents the syntheses of two essential processes: generation and evaluation. And it is only in a fully digital design process that the transformation and the generation of a geometrical model can be supported. It is also only in a fully digital design process that this geometrical model can respond to analytical evaluation, at the same time, and through virtual simulation. It is this holistic integration of evaluative simulation with digital form generation and modification which is at the core of what is generally known as performative design (Oxman, 2006).

This opportunity to fully integrate simulation in the design process provoked a transition from a design process which was traditionally and essentially a process of "form making" to a process which is a combination of "form making" and of "form finding". The new techniques and methods which are applied in contemporary digital processes are much in spirit with the methods and tools pioneered by Frei Otto (Bechthold, 2008), who used soap bubbles for shape optimisation (Fig.4) and Gaudi (Huerta, 2006), who used his famous "hanging models" to explore design solutions for the Sagrada Familia Cathedral in Barcelona (Fig. 5).

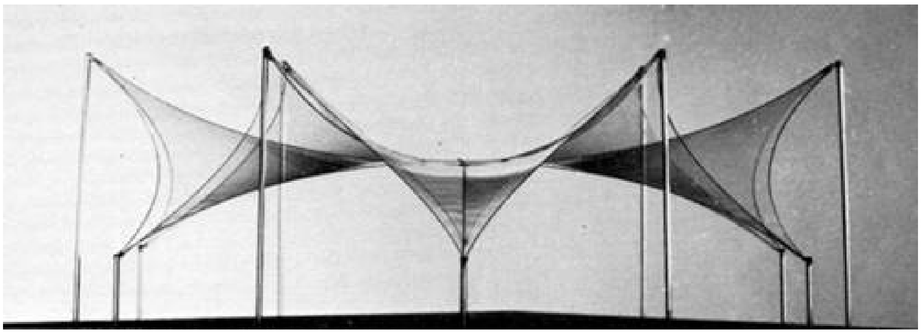


Fig. 4 – Soap Bubble study model for Tanzbrünnen Köln (Photo by Frei Otto).

Fig. 5 – Hanging models by Gaudi (Photo by Tillnm)

Within the framework of a digital design this process of “form finding” is also called “performative morphogenesis” (Oxman, 2008). This is directly related to the notion of morphogenesis such as applied in medicine and which concerns the processes formation and differentiation of dividing cells during embryonic development of an organism. During the last decade the idea of morphogenesis has been the main driver for the development of many architectural projects. Since 2003 Achim Menges and Michael Hensel have been researching the intricate relations between morphology and environment with the development of material systems for form finding coupled to environmental performance. Over the years their students have been experimenting with morphogenetic strategies and material systems that can modulate and in turn be modulated by environmental conditions. They have called their approach “Morpho-Ecologies”, and describe it as a correlation between morphogenesis and ecology, rooted within a biological paradigm, and concerned with issues of higher-level functionality and performance capacity (Hensel & Menges, 2007). In their approach they try to achieve complexity and performative capacity integrating the process of formation and materialisation. Those material systems are used as the main generative drivers in the design process of a complex polymorphic systems based on input and feedback relations. Some of the experiments involve multi-objective form finding, which turns out to be quite complex, but at the same time very interesting. However, these complex material systems cannot always be optimised for e.g. structural performance and minimal material use. So the digital form finding process has to be complemented with physical models at a reduced scale, not only as traditional representational models, but as a full blown and most valuable part of a methodological design process. Scaled functional models have to be built with the capacity of simulation and analysis functions. Rapid prototyping models are used to test geometrical e topological coherence of larger assemblies of elements, and full scale prototypes serve for experimenting with building and manufacturing constraints as well as rigorous testing of real world physical performance behaviour.

Performance based design is also closely related to a specific process of form finding based on the principles of emergence. Emergence has been defined as "the arising of novel and coherent structures, patterns and properties during the process of self-organisation in complex systems" (Goldstein, 1999, p. 49), but in this case emergence has to be understood as a "descriptive term pointing to the patterns, structures, or properties that are exhibited on the macro-level" (Goldstein, 1999, p.58). The concept of emergence has been widely explored in the form finding process in architectural design. It is a process of exploration of the solution space by turning implicit form, explicit. This way it can suggest new forms and possible conceptual directions (Fig. 6). More traditionally, emergence in design has always been linked to creative sketching as a tool or a method to search for new formal relationships.

Fig. 6 – Emergence in Form Finding: Cell House by Tom Wiscombe (Photo by Wiscombe).

Generative design is yet another and different approach in form finding applied specifically in architectural design. Frazer was one of the first architects who applied the concept of generation and he pioneered a design process where architectural form is developed based on code that contains detailed instructions about the generation of the form (Frazer, 2003). The results (Fig. 7) of this process are visual representations which are evaluated on encoded selection criteria. Similar methods, based on this pioneering work are still further researched, refined and adapted. One such example is the research by Janssen (2006) into team based design development of buildings based on an orthogonal grid. With this generative evolutionary design method Janssen demonstrates the design of complex, intelligible and unpredictable three dimensional buildings.

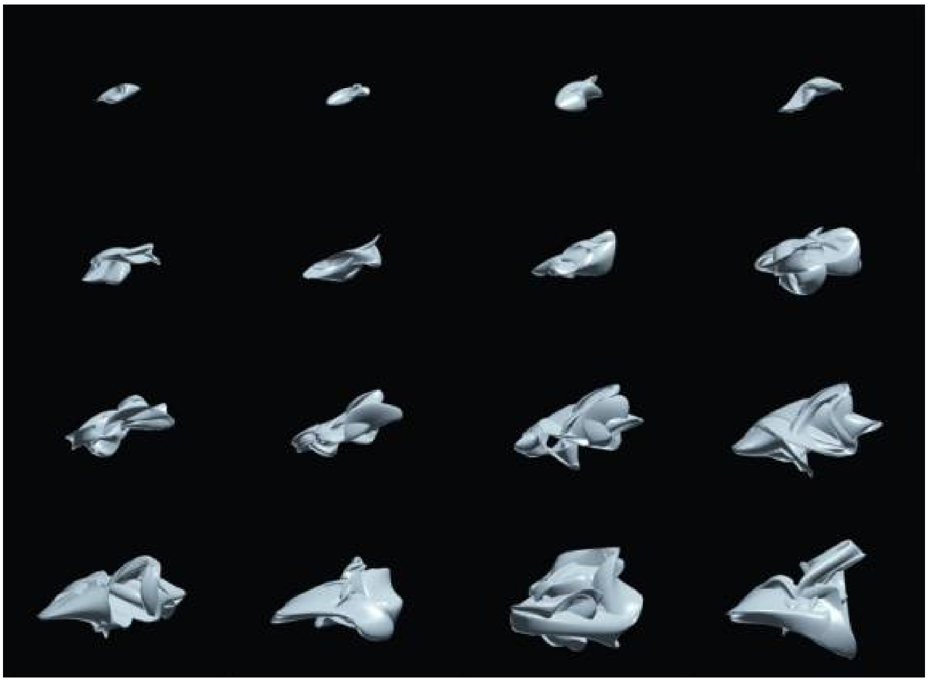


Fig. 7 – Interactivator: Networked Evolutionary Design System by Fraser et al. (AA, London, 1995).

Generative systems are an essential part of the future development of performative architectural systems where evolutionary principals are applied in the initial stages of the design process with the intent to automate and intensify explorative research.

2.3 Design and Optimisation

Ever since the very beginning of the use of computers in the design process, optimisation has been used in every design field with the sole purpose of finding a "best" solution in relation to a set of previously defined performance requirements and optimisation is understood as the process or methodology to make something as perfect, as efficient or effective as possible. This is also how optimisation is defined (Merriam-Webster, 2003) and how it is understood and accepted among researchers in different fields (Rao, 2009). Radford and Gero state:

> (...) Optimisation models effectively search the whole field of feasible solutions and identify those best suited to the designer's stated goals. Thus, optimisation directly approaches an answer to the designer's fundamental question of what is the best solution (1987, p. 25)

Recent development of readily available computer technology and successful research in the mathematical tools and techniques for optimisation allowed for new and different approaches to design. While traditionally design was a cyclical process of analysis, synthesis and evaluation where the designer simultaneously learns about the problem and the range of possible solutions, design by optimisation uses decision making algorithms in order to generate prescriptive information on the nature of an optimal solution satisfying initially specified objectives and within previously specified boundaries. Design by optimisation, as is proposed in this thesis, offers the potential for better design by considering a much broader solution space searching for eventual serendipity.

In the field of Architecture and Building Design pioneering research has been done by Radford and Gero (1987), which resulted in an excellent overview and analysis of different methods and techniques used in the early stages of the use of optimisation in architectural design, illustrated with examples and detailed case studies. They conclude:

> (...) Design optimisation fundamentally involves two issues: how can we do it, and what can we do with the results? (...) how [those results] provide the designer with qualitative and quantitative information, and what supporting information on such factors as sensitivity, stability, and trade-offs is needed (...) Optimisation is not about mathematics and computer programs, although these are necessary tools. It is about striving for the best, about seeking solutions to human needs, about looking for the elusive ideal answers to our problems (Radford and Gero 1987, p. 318-319).

Research in the field of engineering design and more specifically aeronautic engineering has been, and still is, very prosperous with research towards better tools and techniques for optimisation. It is not within the scope of this research to compare and analyse in depth different methods and techniques for optimisation but a good overview of the different methods of optimisation in engineering design and different ways to perform optimisation can be found in a study by Andersson (2001). His research concentrates on the non-derivative methods because they are more suitable for engineering design problems, most of all because those methods are more likely to find a global optima avoiding getting stuck on any local optima (Fig. 8).

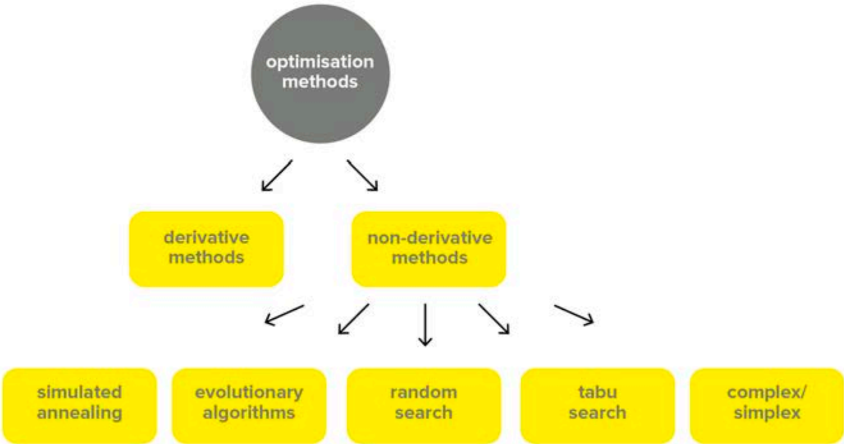


Fig. 8 – Classification of optimisation methods (adapted from Anderson, 2001).

His overview demonstrates how research in optimisation has been concentrated on developing and testing different mathematical models to explore the solution space not only with the objective of finding an optimal solution or making the search for an optimal solution more efficient, but also on how sensitive this optimal performance is to change the design parameters and how robust the decisions are on those parameters which were selected. Besides simulated annealing and evolutionary algorithms, very few of these methods seem to have been applied in research for optimisation in the field of architectural design. Engineering design has a slightly different focus regarding the outcome. An engineering design process tends to be less oriented towards novelty generation and focuses, but instead, more on methods which can support and guarantee a reliable outcome with high quality and competitive costs (Rao, 2009). Accordingly, an engineering design process needs full control over the process as well as over the outcome. In this way an engineering design problem is often resolved by choosing the optimum assembly of standard components in order to arrive to a satisfactory solution (Ullman, 2009). Research in engineering design optimisation has therefore concentrated on procedural methods that must lead to a solution, as opposed to "heuristic methods that rely on a global holistic view of the problem and cannot guarantee that the process will lead to a solution" (Kalay, 2004, p.256), which traditionally is more appropriated for an architectural design process (Tschimmel, 2010b).

Optimisation in engineering design is therefore focussed on the perfect solution (Rao, 2009), but the notion that there is only one single optimal solution to the myriad of goals, aims and ideals which make up a design problem is at least a little bit too simplistic. No solution will be optimised for all design criteria, this is part of the intrinsic characteristics of a design project, and the responsibility of the designer involves the judging and weighting of criteria. Design is about decision making and requires judgement and trade-off's based on the best available information. Therefore the role of optimisation in design is to provide the designer with quantitative and qualitative information. This information is a means for increasing the designer's understanding of the design problem and the nature of good solutions to some of the design objectives. Therefore optimisation in design can also be understood as 'improvement of solutions' or more 'efficiency' instead of optimisation in the strict sense of its definition.

The different methods described in the next paragraphs all have their advantages and shortcomings. In the field of engineering design many optimisation methods have been comparatively studied. Some methods are faster and less computational expensive, but get stuck at a local optimum or can find only one single optimal solution. Other methods are more robust, but need more computer time and are a little bit more difficult to implement. It is therefore obvious that the hybrid combination of different methods could produce quicker or more reliable outcome (Choudhary & Michalek, 2005).

Particularly GA's are many times combined with other more robust methods. A small percentage of a population can be generated by one kind of complex methods and further optimisation can then be executed with the usual genetic operators (Yen, Liao, Bogju Lee, & Randolph, 1998). A possible example could be the use of a method which explores the search space completely and appoints those regions of interest where another method with a higher convergence rate could speed up the process.

2.3.1 Multi-Objective Optimisation and Pareto Optimum

In a design process a designer typically deals with several, often conflicting, objectives. Finding the right solution or a best approach for each of those objectives, let alone for all of them together is not an easy endeavour. Unfortunately most of the traditional and simpler optimisation techniques usually only operate with just one of these objectives at a time. Differential calculus is such a simple way to find a solution to a straightforward design decision problem. Linear programming, nonlinear programming and dynamic programming are other techniques for introducing optimisation strategies in the design process where a single objective can be clearly formulated and understood (Radford & Gero, 1987). Those traditional tools, methods and techniques for optimisation, which are available to assist the designer with optimisation tasks, can only be used to produce additional design information and are most of the time restricted to support limited decision making in the detailing phase of the design process. As such, design as a goal-oriented and decision-making process can rarely benefit from this kind of approach, this is not how a valid novel design solution can be discovered or explored, nor how a design process should or is developed.

Unfortunately, design problems are never simple and straightforward, and so it is difficult to think of any real world design problem which is not characterised by the presence of many conflicting objectives, boundary conditions and requirements. In the design problems a designer gets involved with, especially those problems related with environmental performances that are multi-criteria in nature; he usually does have more than one objective to achieve. It is therefore possible to look at any design problem as an optimisation problem and those optimisation problems are almost always multi-objective optimisation problems. However, different objectives regarding different aspects of performance are most of the times conflicting. Improvement in quality of one of the objectives can reduce the performance of another objective. Optimum performance to one objective usually implies unacceptable low performance of other objectives. Therefore, in multi-objective optimisation problems one single best solution may not exist and a possible solution or solutions are usually a trade-off between conflicting criteria which are difficult to compare. Sometimes objective functions can be optimised separately from each other just to gain some more insight into each performance objective and thus gain additional knowledge about the solution space.

In multi-objective optimization problems there is not one *single* best solution but a population of solutions. This set of solutions can be graphically represented showing a Pareto frontier (Fig. 9) of optimum solutions displaying different trade-offs between conflicting criteria (Ciftcioglu & Bittermann, 2009) (Fontes & Gaspar-Cunha, 2010) (Caldas 2008).

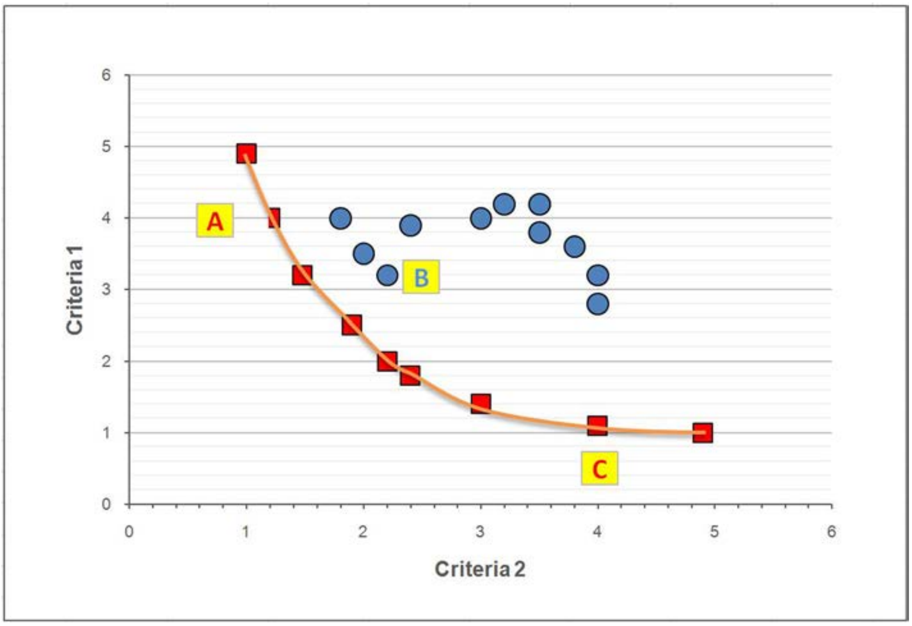


Fig. 9 – Pareto Optimum Frontier formed by the red dots.

Whenever the optimisation problem involves two conflicting objectives, this set of non-dominated solutions can be depicted by a two dimensional "curve", such as the Pareto Optimum Frontier depicted and illustrated by the red dots on the graph in figure 9. For a multi-objective optimisation problem with three different objectives, a Pareto Optimum Frontier can be depicted and described by a curved surface in a three dimensional graph.

Pareto optimality uses the concept of dominated and non-dominated solutions. The result of a multi-objective optimisation process should be a set of non-dominated (Pareto optimal) solutions, this in contrast to a single objective optimisation problem where the result is a single optimum or a set of equivalent optima (Deb, 2001). If between any two solutions none can be considered better than the other on both objectives, these are called *non-dominated* (red dots on the graph in Fig. 9). A solution is Pareto optimal if it is not worse than another solution in all the objectives and better in at least one objective. In other words, a solution is Pareto optimal if it is not dominated by any other solution (Deb, 2001). In Figure 9, for solution B there is a solution A which is better than B for criteria 1 as well as for criteria 2.

Since most of the optimisation problems in design and architectural design are in fact multi-objective and the goal of optimisation in a design process should go beyond the generation of factual information, different methods, tools and algorithms have to be applied. Those methods for building multi-objective optimisation models can be classified in two general approaches (Radford & Gero, 1987). A *non-preference approach* is limited to the production of information on non-dominated performances and on solutions with that performance. The solution chosen by a designer from this Pareto set is often called a 'best compromise' solution because there always has to be a trade-off between decisions that would be in favour of one set criteria and decisions that would point to another set of criteria. In a *preference approach*, the designer's trade-off preferences are placed within the optimisation model. The set of feasible solutions is narrowed down by available information together with rational decisions about preferences by the designer.

A non-preference method (Pareto method) is thus a bottom-up approach that provides a lot of information for the designer to make his decisions. The preference methods are a kind of top-down approach, which provides much less information and attempts to determine in advance *what* information the designer needs to make decisions (Radford & Gero, 1987). In this model and in this thesis both approaches will be balanced according to the design problem which is researched. This way the design process can find a fine balance of potentiality in a digital process where prospective solutions are no longer possible to grasp by human reasoning alone.

2.3.2 Evolutionary Algorithms

Evolutionary Algorithms (EA) are heuristic search methods which have been applied in optimisation problems in a wide range of fields. They have been developed with the goal of presenting a population of optimised solutions instead of just one single point (Marler & Arora, 2004). EA are very robust methods and can handle all type of fitness parameters and variables (Andersson, 2001).

Research on the use of EA for optimisation in the field of design related fields goes back to the early 1970's when Rechenberg and Holland first published their work on this subject. Rechenberg and his research team applied the concept of Genetic Algorithms (GA) for the optimisation of complex engineering problems, and Holland studied adaption and complex adaptive processes providing support for the development of evolutionary algorithms (Holland, 1992).

Evolutionary Algorithms are based on the principals of natural selection (Deb, 2001). Each optimisation parameter is coded into a gene as a string of bits. All optimisation 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). Each individual represents a solution and a set of individuals form a population. Within one population the fittest are selected for combination. The combination of those genes results in a child. The children are reinserted in the population and the procedure starts again. The optimisation continues until the population has converged or until the maximum number of generations has been reached. A lot of different kinds of EA have been developed, all with different features in order to solve a specific type of problems. Especially in the field of soft computing EA's were refined and optimised with the objective to make them more useful for realistic applications. A first practical Pareto based approach to Multi Objective Evolutionary Algoritms (MOEAs) was developed and proposed by Goldberg in 1989 (Fonseca & Fleming, 1995) and this seminal work was the basis for further research in EA and their practical applications (Baeck, Fogel & Michalewicz, 1997).

Important for research in evolutionary algorithms for creative applications is the development of techniques to avoid the tendency to lose diversity within the population of feasible solutions and to converge into a single solution. Therefore the genetic algorithm is modified to function with multiple objectives and applies niching pressure to spread a diverse population along a Pareto optimal trade off frontier or surface (Fonseca & Fleming, 1995) (Caldas, 2005). Those techniques are modelled after the idea of niching in the study of species in nature where natural evolutionary processes maintain a variety of species. Digital evolution research platforms, such as 'Avida', are available to the research community for benchmarking and referencing mathematical and computational applications and algorithms developed to understand the complexity of evolution and propose techniques to avoid the pitfalls of earlier EA's which could not avoid the paths leading to evolutionary dead ends (Avida, 2011).

Thus, niching and other techniques are used to avoid that only one solution is located even when multiple solutions exist. This happens in traditional EA's when individuals in a population become nearly identical too soon. A niching technique allows for EA's to maintain a population of diverse individuals and are capable of locating multiple optimal solutions within a single population. The maintenance of diversity is important because diversity along the Pareto frontier helps in the search for new and improved trade-offs, which at the end are the ultimate goal of the use of optimisation in the design field.

For research in the field of architecture and design a MOEA developed and adapted by Gaspar-Cunha (2009) can be used (Fig. 10). This Reduced Pareto Set Genetic Algorithm (RPSGA) uses the technique of clustering to reduce the number of solutions on the Pareto front, thus contributing to a more efficient process of optimisation and making this particular kind of MOEA a possibly valuable part of interactive optimisation.

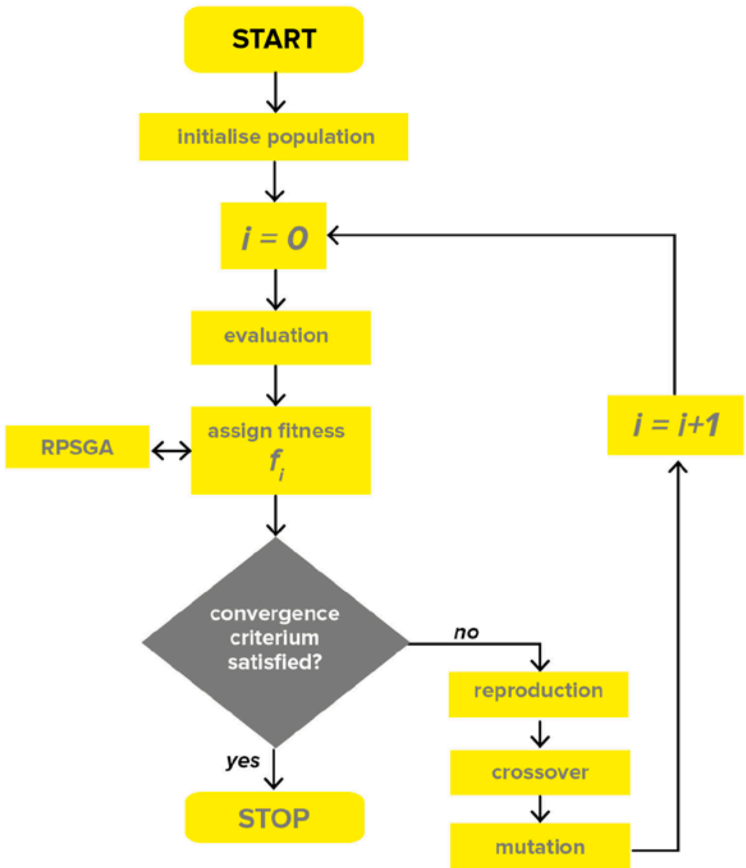


Fig. 10 – Flowchart of a MOEA (adapted from Fontes & Gaspar-Cunha, 2010.

In the field of architectural design, Caldas has done some extensive research in the use of genetic and evolutionary algorithms with the objective of optimisation of multi-criteria problems, involving the improvement of environmental performance in building design. She introduces a Generative Design System as a method that incorporates evolutionary systems and adaptation paradigms in an architectural design process. Her Generative System is based on a Evolutionary Algorithms as search and optimisation engine.

She extends her research to multi-criteria problems using Pareto-based methods to evaluate the generated geometries for conflicting objectives. (Caldas, 2005; 2006) (Caldas and Norford, 2002).

She also studies and introduces the technique of niche induction in the application of GA's in complex domains, compares different approaches and applies her conclusions to the testing of an existing building by Alvaro Siza.

In her final conclusions she raises the question if the integration of all possible evaluation measures of a building in one single system, using a kind of building DNA that would search for the optimal solution for all those evaluation criteria, can or will be desirable. She further concludes that the concept of an optimal solution as the ultimate goal does not make sense in a highly complex domain such as architecture. It might be better to get some insight and understanding in part of the process and leave some other decision-making to the personal interpretation of the designer or the architect. Design intent, she affirms, cannot be excluded from an architectural design process, and design intent depends in part on the designer or the architect himself (Caldas, 2005).

In a different field genetic algorithms were also applied by Eckert (1999a) in the development and testing of a special purpose model for the automated design of knitwear. She argues that interactive generative systems can be powerful tools for human designers and that those systems naturally fit into human design thinking. Her research also indicates that generative tools increase the creativity and the productivity of human designers, and that those generative tools can be used in a variety of design tasks in an easy, intuitive and effective way. However she points out to *human bias* as the main factor which disturbs, and ultimately distorts the objectives of an optimisation process. Although one can argue that it is precisely this bias which makes optimisation acceptable as part of a design process.

Based on the same principles, Eckert and fellow researcher Ian Kelly (Eckert, Kelly, & Stacey, 1999b) have implemented several evolutionary systems to assist artists and designers in selecting colour combinations. Kelly's aim was to develop a generic tool that exploits the findings of colour science and helps the designer with the selection of colours.

2.3.3 Simulated annealing

Simulated annealing (SA) is a generic meta-heuristic based on research in optimisation of complex systems (Kirkpatrick, Gelatt, & Vecchi, 1983). Simulated annealing is inspired on the analogy with the annealing of solids in metallurgy. Annealing in metallurgy is a technique which involves heating and controlled cooling to manipulate the size and configuration of crystals. In an optimisation problem the SA algorithm replaces each current solution with a nearby random solution and uses specific mathematical techniques to avoid that the system gets stuck at a local minimum, missing out possible good solutions. Simulated annealing is a robust method and less computational intensive as compared with genetic algorithms.

Shape grammars and simulated annealing have been used by Kristina Shea as a basis for a computer construct for topological optimisation. She actually developed eiFORM, which she describes as a software demonstrator for generative structural design and optimisation based on a method called Structural Topology and Shape Annealing – STSA (Shea, Aish, & Gourtovaia, 2005). EiFORM is a generative method that combines grammatical parametric shape generation, structural analysis routines, performance evaluation and stochastic optimisation to support optimally directed exploration of discrete structural forms. Using this method she has generated multiple design alternatives for planar truss structures (Shea & Cagan, 1999), transmission tower design (Shea & Smith, 2006) and canopy structures built at the Hylo-Morphic Project in Los Angeles in 2006 (Fig. 11).

Fig. 11 – The Hylo-Morphic Project in collaboration with Kristina Shea (Schindler House MAK LA, 2006).

2.3.4 Artificial Intelligence and Automated Design

Building a machine or a software application which can interact with humans intelligently and which can solve creative problems autonomously has been the dream and the main objective of many researchers, even before the advent of the computer. If this could ever be achieved one would have a fully automated design process without any flaws and with guaranteed success.

The hope to be able to design fully automated design systems was directly stimulated by the design methods movement and by the introduction of the computer in the design process. The goal of this research in Artificial Intelligence techniques was not only to take over boring routine design from the human designer, but produce design proposals completely different from what would or could be expected from human designers. Decades of research, however, did not lead to any kind of *Universal Design Solver*, although a lot of interesting work has been proposed based on basic research in Artificial Intelligence (AI). Many research papers have been dedicated to this research topic, but besides the use of rule based systems and some sporadic application of neural networks little of this AI based research has found practical applications in the field of product design or architecture (Gero, 2007).

Only in fields such as electronics and software design, some researchers have tried to build expert systems and knowledge-based systems that can learn automatically based on machine learning techniques (Kumar, Subramanian & Teck, 2000). While other research has tried to improve the quality of decision making by developing mixed-interaction systems based on coordination between people and fully autonomous agents (Peng & Gero, 2007a). But besides positive results in those very specific situations, no research along this line has been conducted in more creative fields such as architectural design or product design.

3 Concept Formulation and Implementation

The principal aim and objective of this study is to integrate the employment of simulation software and optimisation techniques in the design process within a framework of design as information processing. The method presented in this thesis also aims to provide the conceptual tools and guidelines for designers to organise and structure a personalised and goal specific digital design process, and do so effectively and efficiently. The proposed structure should offer a body of meaningful descriptions and rules to build proprietary applications for conceptual design.

The development of this research project is based on the presumption that initiatives to support and enable decision making in design lead to better designing, and thus better solutions. Axel Kilian (2006) defines contemporary design as a dynamic process of generation, emergence and discovery, and recognises the ability to explore the relationship between the initial definition of requirements, and the relationship between those requirements and the constraints of the project as the most important contribution to the quality outcome of a design process and ultimately to better adapted design and architecture.

The means to determine and refine those relationships between requirements and constraints is based on the use of mostly prescriptive simulation and evaluation software. The main concern of these software applications is the presumption of the existence of an objective reality. Methodological processes, formal specifications and rational reasoning are favoured over subjective viewpoints, individual intuition and ambiguity, and uncertainty. A functionalist inside-out approach, where shape derives directly from the constraints, dominates in engineering design for complex products (Culha, 2005). This positivist attitude towards a design problem, based on rational problem solving, is very common in straightforward engineering design, but has been almost completely abandoned in product design or in architectural design. However, as argued by Dorst (2006) positivist or objective attitudes may be appropriate at certain parts or stages of the design process where ideas have to be implemented and presented, while a more subjective attitude should prevail during those ill-structured stages of the design process concentrated on the generation of novelty. The methodological model presented in this study will be part prescriptive and incorporate the maximum of rational approach through the use of methodological processes, as has been argued by Dorst.

3.1 Concepts for the FFO Method

For the proposed digital design method proposed in this thesis, digital design thinking makes an important contribution. Digital design thinking can be formulated as a constructed relationship between information and forms of representation that support design in a computational environment (Oxman, 2006). As such, digital design thinking challenges the fundamental concepts traditionally related to design methodology, such as representation, generation and iteration. Oxman states that we no longer represent form and shape in the traditional paper-based sense, and allow for a new approach, introducing new concepts of dynamic and responsive forms and spaces. But, digital design should not be considered as design with the use of a new set of digital tools. Digital design should evolve from computer *aided* design (CAD or CAAD) to computer *based* design (Kolarevic, 2003).

As a conceptual framework for digital design based on methodological characteristics, Oxman (2006) proposes a compound model (Fig. 12) of digital design as the final model of evolution in digital design after a dual-CAD model, formation, generation, performance, performance-based formation and performance-based generation. Her compound model is based on the integration of processes of formation, generation, evaluation and performance. It demonstrates the growing sophistication of digital design media and its impact on the design process.

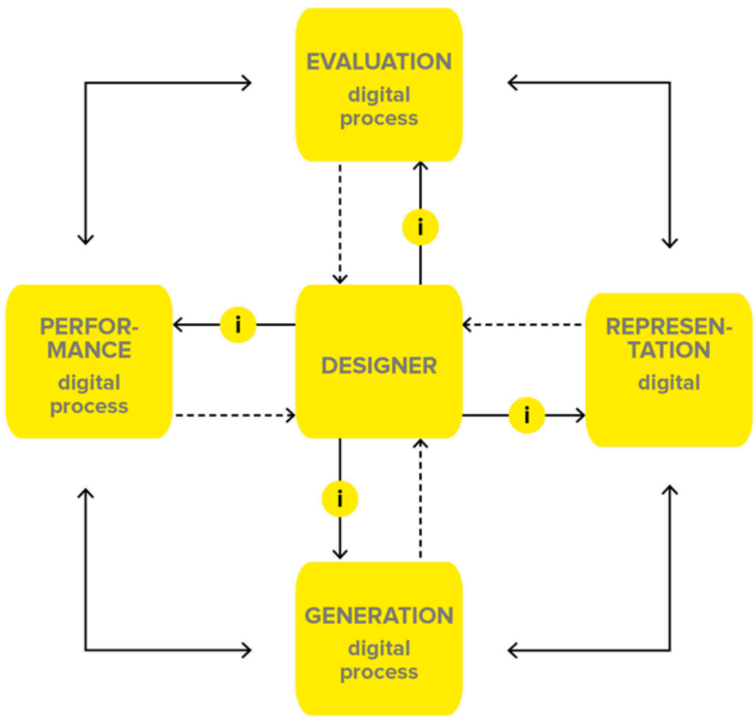


Fig. 12 – Oxman's Compound Model (adapted from Oxman, 2006).

Within a compound model of digital design, the designer has a new role according to the nature of his interaction with the digital media. The digital designer interacts, controls and moderates generative and performative processes. The designer manipulates information, this way the traditional role of the designer-as-user of tools changes into the designer-as-toolmaker.

The model proposed in this thesis is intended not only to assist the designer but also to collaborate with the designer both by evaluating and by proposing possible solutions within a delineated area of an overall solution space. Optimisation in this process will be used in the perspective of a negotiation process between the forms envisioned by the designer and the information generated by the simulation (Kolarevic, 2005). As such, topology and geometry should not be fixed but open to modification. In this process design shifts from pure modelling to defining principals and systems with a specific behaviour. The resulting shape emerges out of a process of exploration and careful balancing of multiple objectives. Kolarevic concludes that in such a process the emphasis shifts from "making form" to "finding form" (Kolarevic, 2005).

Digital design systems contain three main components that should be integrated in a holistic design process: a geometric model, an evaluative process and a kind of interactivity between the designer and the computer. Integration must be understood in the sense of interdisciplinary integration of different tools and methodologies and must be approached with the objective of achieving a better design.

The geometric model has to be formulated in such a manner that it is capable of producing interesting results and make generation and transformation possible and useful for evolutionary algorithms. Usually this geometric model will be built parametrically or will be adapted with a kind of parametric behaviour in a CAD environment. This geometric model should be built in such a manner that it can respond not only directly and automatically to the outcome of independent simulations and evaluations, but also to direct input from the designer.

The evaluative processes also have to be fully integrated in the design process. Mostly this can be achieved through the coupling of simulation or analysing software directly with the CAD software application which can change the existing geometric configuration of the model or which can eventually generate a complete new geometric model. But to aim for a truly interactive optimisation method, all the sub-processes and systems involved in this design method have to be open and accessible to the designer. The software construct cannot be configured as an independent and fully automatic design process, or it would evolve in a closed design tool. The designer still has to be the moderator of the various processes involved, and outcome has to depend solely on his judgement.

In the next paragraphs the general requirements and objectives for the development of the proposed design model or method are summarised. These requirements will be used to develop the concept for a modular digital optimisation model.

3.1.1 The Designer as Toolmaker

Tools are artefacts made by man, applied to certain actions and with the objective to attain specific goals. The function of a tool is to enable its user performing tasks (Merriam-Webster, 2003). Tools are made or constructed to perform in certain ways and in design literature these ways are usually referred to as methods (Fisher, 2008).

He also concludes that design tools and particularly digital design tools are by nature restrictive in use and limited by the objectives of the developer of the tool, usually a programmer or code designer, but almost never an experienced designer or architect. Therefore many of those tools offer limited and predefined possibilities and unless they are used by the designer with other purposes than for which those software tools were originally developed - this kind of use is sometimes referred to as hacking - they do not tend to contribute to a more creative outcome. Therefore some researchers have also suggested that CAD and CAAD tools can be more useful to designers if they can be constructed and built by the proper designers instead of relying on the thinking process of those software designers who originally conceived the software application (Gero & Kelly, 2008).

The tools applied in design can broadly be divided in two categories: one collection of tools, usually non-computational but not necessarily non-computer based - the same graphical presentations can either be paper based or represented on computer screens - are focused on providing stimuli guiding creative design exploration. Another kind of tools are usually exclusively computer (software) based, and are primarily focused on the development of solutions (Fisher, 2008).

Some researchers have constructed closed systems tools aimed for the design of very specific objects. Intelligent Genetic Design Tool (IGDT) is such a tool specifically developed for the design and optimisation of architectural trussed structured systems (von Bulow, 2007). In his research work this author proposes a new class of computational tools aimed at intelligent interaction with the designer in the conceptual design phase. Very similar to the approach in this thesis, his tool focuses on exploration of proposals and offers multiple good solutions as a result, this way avoiding fixation on a single best optimised solution. His approach to optimisation allows for the integration of hard to code criteria such as aesthetics and meaning. The concept of his tool expects human interaction at all levels with the goal of stimulating the designer to discover creative solutions. However, it is a tool, a closed system, and thus it can only be used for the exploration and optimisation of a specific kind of trussed structural systems, and presents results which belong to the same expected type or group. According to the research of Fisher it does not seem to be possible to make a kind of universal software tool "outside applied projects" (Fisher, 2008, p. 210). Unlike traditional tools such as a hammer or a screwdriver, a software tool always has to be adaptable to the particular situation where it will be used. And he points to another characteristic of these software tools, the process of tool-making is self-referential: new "tools are developed based on tools developed previously by others" (Fisher, 2008, p. 174), existing tools are modified or their use is deliberately subverted from the way of use originally intended by their makers. Therefore one can no longer speak of a tool but have to call it a method or a system.

The basis for the selection of the required tools in the proposed method depends on design goals and the design requirements. Some of those requirements or constraints remain unchanged throughout the design process and are the result of the initial design intent of the designer. Those requirements are the basis for the selection of the different design software programs which will be part of the proposed optimisation method.

Glanville (Glanville, 1994) makes a distinction between tools and medium. The computer used as a tool follows instructions from the user according to intentions preconceived by the maker of the computer tool. The output produced is "expected" by the designer and directly related to the input. Using the computer as a medium produces results which were not foreseen or expected neither by the designer nor by the toolmaker, this way one can speak of interaction between the designer and the computer. This is very important for the mind-set of the designer: the perception of having control over specific software application, e.g. the simulation model, is crucial for the acceptance of the design tool. According to Roberts and Marsh (2001) the automation of the data input has to be implemented very carefully so that the modelling is perceived as transparent and logical. It is the designer toolmaker who has to decide between what needs to be calculated and what can be evaluated or estimated by intuition. And it is the designer toolmaker who selects the kind of design tools which must be integrated into the design process and for which activities of designing, and which possible different levels of design specification will be used to generate useful information.

If the proposed software construct should act as a closed system, similar as existing CAD software, limitations on the outcome could seriously hinder the flexibility of the design process. Choosing and selecting adequate software applications therefore seems to be the right option for the introduction of optimisation into the design process. Only this way a robust design explorer can be constructed which is fully adapted to design intent *and* to project characteristics.

3.1.2 Design Strategies

Important research has been done to analyse and compare different design strategies. Protocol data were used to analyse singular cognitive strategies employed by designers (Kruger & Cross, 2006). The outcome of the design projects were classified by experts, compared and analysed. Based on these results and information from protocol data, four distinctive design strategies were identified by Kruger and Cross: knowledge based design, information driven design, problem driven design, and solution driven design.

Knowledge based design - where the designer takes the knowledge he already has as the basis for proceeding - and information driven design - where the designer spends most of the time gathering information - are not those strategies which proved to result in the most creative solutions. Problem driven design is the strategy where the designer focuses on the problem, and uses almost exclusively information and knowledge that is needed to define and solve the problem as quick as possible. This strategy results in either a highly defined problem which leaves little room for solution alternatives or a rather abstract defined problem. In both cases the results are strongly focused.

The designers who used this strategy, use knowledge about structuring and refining problems, most of the time based on knowledge of former related cases. This kind of design can almost always be defined as re-design.

Solution driven design, on the other hand, is characterised by a short problem analysis stage followed by a long generating and evaluating stage. Typically a large number of solutions are generated, possibly more varied in quality. The designer uses the solution to further define the design problem. Sometimes the problem is even reframed only to justify an interesting solution.

As could be expected, no expert designer uses exclusively only one of these strategies, but mixes different strategies with different emphases and intensity according to the type of design process in which he is involved. Kruger and Cross' research project demonstrates that the best results in almost all assessed solution aspects, including overall solution quality, were obtained by the designers who employed a problem driven design strategy, better even if combined with a partly information driven strategy. The most creative solutions were obtained by those designers who used mainly a solution driven strategy. This might imply that, independently of the design-strategic orientation of the designer, a design method or a design tool for designer-computer synergies should aim for the possible integration of multiple strategies simultaneously.

The proposed method for introducing optimisation in the design process does aim to combine and integrate multiple design strategies. In order to be able to build the software construct, the designer has to fully understand the design problem and carefully select and define relevant parameters, constraints and boundary conditions. These actions, characteristically for a problem driven design strategy are the pillars for building a functional software construct for the introduction of optimisation in the design process.

Knowledge driven design and information driven design - although Kruger and Cross (2006) classify them as distinctive strategies and more or less as opposed to solution driven design - are important for building the software construct. Which software programs to choose and how to connect them is important to certify that the right information is exchanged, retrieved or communicated to the designer. Also the correct application of the simulation software relies on some expertise and experience in its use. Furthermore, both strategies are also the heavily influential on which kind of geometry and how it will be constructed, in order to guarantee fluent development of optimised results according overall design intent.

But the overall strategy supporting the correct application of the optimisation method, and thus the building of the software construct is solution driven. All stages in the optimisation method are characterised and focussed on iterative feedback with the purpose of generating qualitatively varied solutions and providing the designer or architect with enough information for rational and emotional decision making or possible reframing of the initial design problem.

3.1.3 Conceptual Design

Conceptual design is very different from the other phases of the design process it is difficult to structure and to control. It is the phase where the designer has the most freedom and where constraints and parameters are less important and can be tested or discarded. It is the phase of dynamic exploration and creative stimulation (Benami & Jin, 2002) (Liu, Chakrabarti, & Bligh, 2003).

Providing overall computational support for the design process is difficult and none of the existing CAD design tools offer adequate support throughout the complete design process. And although many digital design environments and tools were developed to support particular activities during certain specific parts of the design process, almost none are targeted to the design activities in the conceptual phase of the design process (Horváth, 2005) (Chong, Chen, & Leong, 2008), or are very restrictive in application range (Gupta & Okudan, 2008).

How to structure the conceptual design phase, and if digital tools or methods can contribute to a more fluid design process, remains very problematic and controversial, and many designers are still highly sceptical toward the value of computer assistance in conceptual design. In some educational programs in architecture or product design the students are discouraged or even forbidden to use the computer, especially in the early stages of the design process, and sometimes throughout the complete development of their projects in the first year.

In the beginning of a design process, when the basic characteristics of a project are defined, designing is relatively unstructured. The designer relies on very quick feedback of sketches, digital sketches, coarse models and basic renders (Jonson, 2005). The emphasis is most of the time focused on visualisation rather than precise modelling and detailing. The detailed study and technical development of a concept is a more structured activity involving the quantitative and qualitative development of the design intention. This activity, also sometimes called embodiment design, is usually executed in the 'detailing phase' of the design project (Pahl, et al. 2006).

The emphasis in conceptual design is invariably on initial shape generation. Any tool used during this stage of the design process has to act as a kind of visual thinking tool for the designer, and has to be a stimulator for the designer´s own creativity. Decisions in the early stages of the design process determine the potential performance of the design objects later in the real world. Ineffective and inefficient processes can lead to mistaken decisions which can be difficult or impossible to correct at later stages.

Also, in the conceptual stage the designer mostly works with belief and imagination. It is the objective of the method presented in this study to assist this assumption and imagination. This process is important because the conceptual phase of design is also a "learning phase". During the conceptual phase the designer will always generate new proposals in iterative circles and incorporate knowledge gained from a previous proposal into the creation of another proposal. This iterative process induces the designer in a better understanding of the design problem and informs about the necessity for reformulation of requirements and boundary conditions, or incorporation of new or complementary specifications.

Possible pitfalls for the creative outcome of a design process are the consequences of a combination of analysis and simulation software applications. These results can push the designer for the acceptance of one kind of solution, and this can be considered an imposed limit to creativity simple by the application of too much information in the early stages of the design process. It is generally stated that new solutions are more likely to emerge from less detailed design representations. Too much information, too early, can diminish creativity. It is therefore necessary that the proposed method presents many possible and different solutions which guide the designer in the right direction avoiding exploring solutions with less performative qualities. Eventually the design strategy has to be reviewed and reformulated, and a new iterative set of cycles started. Conceptual design is and should be free and very dynamic.

3.1.4 Design Explorer

Designs are created for a purpose and this purpose is what initiates a design process. This may seem evident, but discovering solutions merely by change cannot be considered design. Building a shelter for safety is design, stumbling onto a cave is not (Lawson, 2005). It is this understanding of design as a purposeful activity what is part of the core of the general accepted definition of what design is. Design with intent does not only imply clearly formulated objectives and carefully planned actions, but appoints also to a state of mind (Merriam-Webster, 2003).

Close to the notion of design intent one can identify another element of a purpose driven design process, the design driver. This concept has been defined as that condition or that constraint with most weight in a design exploration (Kilian, 2006). Many constraints or parameters are and can be defined at the beginning of a design process, and many of those constraints can be refined, adjusted or even abandoned during the evolution of the process. But a design driver is not easily changed and therefore has the strongest influence for directing design strategy and the kind of design exploration which can be applied to the particular process.

The concept of design intent together with the existence or formulation of a design driver are the key features of design exploration, which is a central aspect of act of designing and is paramount to the principal of a solution driven design process. The construction of a design explorer is the basis for complying to design intent. The process of defining boundaries and constraints are itself part of the design process and help to define the problem itself. The selection of requirements and their boundary conditions will be the first act of design.

Any software construct with the intent to act as a design explorer should be able to act spontaneously and be able to return unexpected results. It should be possible to adapt the software construct to different needs and change its structure interactively switching or reorganising its different components. An efficient design explorer also provides for refocusing its functionality and allows for the modification of conceptual models within and around the design task at hand. Important features for such a software construct are the interfaces and the links between all the different modules, so that information can be exchanged flawlessly and without losing details or changing configurations. This is very important as design exploration depends in a great part on profound understanding of the relationships between design constraints. This way exploration in a digital design process can be a much more *process oriented* interaction with the digital media than a *skill based interaction*, such as happens in commercially available CAD software applications. Furthermore, this model for the software construct has to be understood within the framework of integrative design exploration. With integrative design exploration the search process does not focus exclusively on the final form, based on, for example structural analysis, but rather as an integrative method with the design explorer modelled to be able to respond to multiple constraints and requirements. The ultimate goal of such a design explorer is providing a base for rational decision making.

3.1.5 Functional Requirements, Parameters and Constraints

In a conventional approach, design is regarded as problem solving by pure analysis. This view is based on the contemplation that a design problem is well described at the beginning of the design process (Gedenryd, 2008). And whenever this problem is not explicitly stated, then most of the time it is assumed that one can readily specify what the problem consists of. This would be correct if design fits problem solving theory, but in reality, producing or constructing the problem is part of the design process and is indeed one of *the* most important and difficult tasks of the designer.

In this same conventional approach, constraints are considered as a crucial and very important part of the problem definition. Those constraints are given to the designer at the very start of the design process and are listed in the initial design problem description under requirements or specifications, which have to be complied to. Constraints are thus those requirements that point to restrictions on what could be considered as acceptable solutions. These specifications are supposed to make the design task harder by placing strict limitations on the scope of possible solutions. However, only a specific kind of constraint has effective restrictive force, legally imposed building regulations and ISO safety standards in industry are examples of such constraints. This kind of constraint is absolute and beyond the designer's influence, and cannot be avoided because they do not originate from the requirements specifications and are not project specific. For the designer there is no other option than complying.

For most of the constraints, however, this is not quite that clear and straightforward. In reality constraints can be both restraining, and practical or helpful. It is a fact that designers bend, alter, add and change constraints throughout the design process and that this is considered as part of creative problem solving. Furthermore, a specific kind of constraints is usually imposed by the designer at the beginning, or even during the design process. And those are completely flexible and in complete control of the designer because it is he who formulates them. They are adaptable and the designer can take a completely pragmatic attitude towards them. The designer can choose.

Protocol studies show that designers frequently impose constraints that are neither necessary nor objectively valid, but are inherently useful to probe for possible solutions (Christiaans, 1992). How those constraints are used and how they can be a driving force for creative solutions can make the difference between a novice designer and a professional designer. The reason why a designer will add constraints to the requirements specifications are twofold. The professional designer will use his experience as a major justification for imposing particular limitations to help him draw upon personal knowledge to structure his design problem. This way he will be able to apply techniques he is already familiar with and support a divergent way of thinking and exploring. This allows for a "working forwards approach" focused on objectives, on solutions. A novice designer on the other hand will use constraints to support his convergent way of thinking. His blind compliance to constraints will force him to use a "working backwards approach" to design. His design process is problem oriented, deductive and not creative.

Constraints and how constraints are manipulated in this research project is very important. Constraints, and especially a particular kind of constraints called "boundary conditions", are closely related to optimisation. Constraints help to focus on design exploration. As discussed before, choosing the functional requirements and the right constraints at the beginning of our design project has to be considered as an important and valuable part of the design process. The definition of each of the constraints and the possible interaction between them, are themselves already a major part of the design exploration. As such, an important issue in interactive optimisation is the possibility to allow constraints to become design drivers in the exploration of a design problem. This means that on the one hand we have to formulate functional constraints or boundary conditions that provide us with a control principle which will allow for evaluation of the information generated through the different optimisation algorithms. Due to the nature of the mathematical principles which guide our optimisation routines, this part of the design process has necessarily to be formulated as pure problem solving, including given constraints at the beginning of at least this iterative loop. But as discussed before, a design process does not evolve along this kind of rigid sequences. Interpretation and weighting of the constraints has to be part of the articulation of the design intention.

A separate kind of constraints do not have any relation to the design problem or the design solution, but is a result of system and process options and is related to the specific characteristics of the software construct which the designer will build to explore the design problem. This research project envisions the direct coupling between digital generation of modelling information and digital fabrication techniques. Depending on which modelling software application and which kind of digital fabrication technique will be used, specific process constraints have to be accounted for.

Constraints and constraints management in the design process are an important feature of the design discipline itself and have been studied and researched extensively in the field of design (Kilian, 2006) as well as in the field of architectural design (Gross, 1986). According to Kilian form emerges from the "interplay between design intention and design constraints" (Kilian. 2006, p.285) if it is possible to adjust the boundary conditions at any given moment during the iteration process. Even if a design proposal fulfils all the initial requirements and constraints, it may still not fit the design intent. Contrary to conceptual variations, parametric variations can only cover the small area of the solution space. In order to obtain feasible results it is therefore important that the constraints and the requirements are as flexible as possible, and that the constraints are imposed by the proper designer.

An important first step in a design process in general, but an essential step in a digitally mediated design process, is identifying the design problem and its constraints and finding an appropriate way to implement and comply to the constraints. Therefore an initial analysis of the design problem is a very significant and paramount for setting up a parametric geometry with the objective of creating novel solutions. However, one should be aware that parametric variations are but one possible strategy for exploring a rather limited area of the solution space.

3.1.6 Iteration

The dominant approach in Design thinking in Architecture is still largely non-iterative. Architectural design developed from building is historically closely linked to the building process. The architect was the master builder, and the design process was his mental process for producing a building. It remains difficult to imaging and to understand the meaning of a building in reality. In industrial design, thinking processes are mostly non-linear but iterative (Kumar, 2006). The design process is not a linear process from rough sketch to detailed design. Design projects are developed and explored iteratively in a holistic and connected understanding of the design problem, as a continuous cycling between generating and evaluating ideas (Wynn, Eckert, & Clarkson, 2007). The design process is typically a sequence of modifications and extensions. Designers alter previous designs and reuse components and solutions to solve problems. Evaluating decisions through iterative interactions between design and analysis is common practise nowadays among building design and consulting teams.

The recent development of more sophisticated digital design tools for simulation, representation and analysing virtual design proposals, offers unique possibilities to add rigour to the process of formalising decisions based on iterative interactions between design and analysis. This way, a design process becomes a practice that balancing factors and constraints in order to make decisions about how to change things when even basic design rules may change constantly during the process.

The design process is intrinsically iterative because design problems need to be defined more clearly and the problem solving approach must be adapted through repetition (Dominick et al., 2000). These authors describe design problems as open-ended and the task of the designer is to identify an optimal solution amongst multiple alternatives through systematic analyses. Therefore, creating variations is fundamental in the search for solutions of a design problem. Working in an environment which stimulates and supports variation improves the quality of the design process and the quality of the design outcome. A search process of iteration between variations of a design idea is crucial.

Another aspect of an iterative process is the generation of knowledge. Knowledge generation relates to iteration through the observation that knowledge about the design problem grows during the design process. New knowledge, especially new basic knowledge about the design problem, should be allowed to enter the design process at any moment. Since designing is not limited to objective problem solving, it does not involve only search, but involves also reformulations of the search space. All iterations are thus affecting the complete creative ongoing process.

The design model proposed in this research is built around an iterative process where the design involves the constant development and refinement of requirements, the synthesis of intermediate design solutions and the emergence of new concepts from what has already been partially designed. Greater iteration cycles do also need to be possible to allow for more profound alterations to the initial design object and the design goals.

3.1.7 Evaluation, Appraisal and Coherence

The positivist world view and hence the view of design as problem solving, dominant in the research around artificial intelligence and automated design, assumes the existence of an objective reality (Gero, 2007). According to this line of thought, methodological processes, strict formal specifications and pure rational evaluation are necessary for efficient and thorough exploration of the solution space. During the design process individual interpretations based on knew knowledge should be avoided and rephrasing of initial requirements ruled out. But design or architectural design is not about finding *the* 'optimal' solution in terms of a set of criteria: it is also to a great extent an aesthetical endeavour. It is therefore more interesting to move away from 'optimisation' in the strict sense of problem solving and consider the model as a process of adaption. Much in the same way as the approach to optimisation in design, as it has been proposed by Caldas (2006). In such a process, many different solutions are or can be considered equally well-adapted, and the most 'optimised' proposed solution is not necessarily the most beautiful or the most wanted by the designer. Some interference from the designer's preferences is therefore welcome and an utmost necessity.

Humans are good and fast at making perceptual evaluations of complex and subtle properties of designs just by looking at a representation of the object. Training and practice enhance those skills (Eckert, Kelly, & Stacey, 1999). Professional designers develop skills for what has been called perceiving. An expert designer can perceptually recognise features and properties, and evaluate technical and aesthetical quality of a design only by seeing a representation or even only by imagination (Schon, 1991). Designers use this tacit knowledge to recognise which aspects of design are right or wrong. This way, they can have a feeling about what can be the right direction towards a successful design. Many studies have been made about how designers use external representations (Schon & Wiggins, 1992) (Goldschmidt & Smolkov, 2006) (Bilda & Gero, 2008). As a general conclusion of all those contributions, it can be stated that design is a constant process of iteration and decision loops. The research shows that designers make a small proposal, evaluate what has been produced and reformulate the initial design statement with added information obtained by this evaluation. Visual displays of representations of the design play an important role in a designer's creative process. Research on sketching has demonstrated that designers externalise ideas as part of their creative thinking and that sketches are used to enable perceptual evaluations. Sketches are also supposed to activate knowledge held in long term memory as a means to inspire for novel design ideas (Eckert, Kelly, & Stacey, 1999).

According to the study of the neurologist António Damásio (2006) emotions are the basis of everything we think and the human brain is quite prepared to solve problems with great complexity with the help of emotional decisions. This emotional feeling or evaluation is holistic: apparently the brain can process much more information on a subconscious level than it does on a conscious level. Consequently one makes intuitive and emotional decisions before trying to rationalise those decisions on a conscious level and engage in purposeful thought (Burnette, 2009). Rational thinking reduces the complex and broad spectrum of relevant factors to a few manageable ones, without considering the “complex interplay of meaning” in design (Gänshirt, 2007, p.77).

In the design process different levels of action are executed simultaneously. This makes the process very difficult to analyse (Dorst, 2003). Design has at the same time something intuitive, often mystified and purely emotional. But it is also a chronologically ordered process tied to time. Both approaches are contradicting but remain unsatisfactory in their own right. A combination of those two approaches sees design as a cycle of recurring steps. An interplay of seeing, thinking and doing on any different level during the design project is the basis and the justification of a process of constant iteration.

As the computational capabilities are increasing, the usage of optimisation in design is getting larger. What nowadays takes seconds to calculate on an average computer was much more complicated and cumbersome a decade ago. The applications for numerical optimisation and the use of those applications in the design process have increased dramatically. But although those techniques could be a valuable assistance during the development of a design project and permit vast improvements, most of the important decisions are still based on intuition and are made by the designer. Humans are good at perceptual evaluation of criteria that are very difficult to program. This human fitness evaluation has to be used to support validation of the computer optimised generative results. Most of the research which tries to automate qualitative interpretations and systemises appraisal in design does so on a very limited scale and only with very well documented interpretations. The method proposed in this thesis will combine quantitative evaluation with qualitative appraisal, and although some aspects of the design proposal will have priority over the others, all have to be assisted at the same time. In such a process the task of the designer calls for judgement and for problem solving at the same time. How we decide what is good and what is desirable is usually rather intuitive, and different persons might have different perceptions or assumptions about the value of the solution at the early stages of the design process. Furthermore, besides value, a design also needs to be coherent. Coherence describes to what extent a design can be perceived as a whole and without contradictions (Dorst, 2006). The need of coherence of the final design is absolutely necessary in any process of multi-objective optimisation. The amount of compromises the optimisation model can use and tolerate can only be accessed (in this present software construct) by the judgement of the designer. And although one can expect this judgement to be highly subjective, research has shown that the choices made by trained professionals do not defer that much from one another (Christiaans, 1992).

3.2 Implementation of the FFO Method

Unfortunately there is no single digital design environment that comprehensively addresses all needs in relation to analysing, simulating and designing all information necessary to build an object. It is therefore the task of the designer to build his own software construct with the intent to introduce optimisation efficiently and goal-oriented into the design process.

In the following paragraphs the proposed method will be deconstructed and each part or component will be discussed in detail.

3.2.1 Software Construct and Scripting

Since 1994 the International Alliance for Interoperability is working on the development of a standard exchange format for open product data. Its mission is to specify the so called Industry Foundation Classes (IFC) as a universal language to improve the communication, productivity, delivery time, cost, and quality throughout the design, construction, operation and maintenance life cycle of buildings, and to produce a standard for communication (Buildingsmart 2011) (AWCI, 2011). Such a standard communication format would indeed make the development of the kind of software constructs which were tested in this research work a lot easier. Preparing different models in different environments with the goal of not losing any valuable information through translation between different file formats (for example, between CAD tools such as Rhino 3D and simulation programs such as Autodesk Ecotect 2011), takes time, and is not essential to the design process. In the proposed software construct some of the proposed geometries were developed in surface modelling CAD software application (Rhino 3D) which was then exported in a file format which could be imported in the simulation software application Autodesk Ecotect 2011. Some of the necessary virtual models were directly constructed in the native 3D modeller of the simulation software program.

There are many different ways to use software applications in the design process, but for the process to be a truly digital process, not only one software application can be used. A possible and practical solution is the coupling of CAD software application with different simulation and evaluation software. This way will allow the designer freedom to customise according to requirements proposed by the preferred design strategy.

It is important to notice that most of the CAD and CAAD software applications are developed not by designers or architects but by programmers. Unfortunately this seems to constrain and limit the creative and exploitative capabilities of the designer or architect using this software application. In order to circumvent these possible limitations, macros and customary scripting can be used. Scripting is a reasonably simple tool which allows the designer to model at some extent the software application to his own design strategy. It does not require intensive training and allows creative exploration, even with only basic knowledge.

3.2.2 Outline of the Software Construct

The conceptual phase of the design process is part of the design process where production, integration and communication between ideas and knowledge are most important (Fig.13). It is in this phase of the project that constraints, parameters and boundary conditions are specified and possible solutions are generated. A great number of ideas are generated, considered and evaluated, and eventually one of the ideas will be developed in more detail. It is important to acknowledge that the quality of the process of development depends, on a considerable part, on the seamless integration of knowledge in this phase. And although it is possible that an interesting concept results in a poor final proposal, generally good concepts are more easily transformed in high quality final products (Wang, 2002) and poor concepts take a lot of effort and time in the detailing phase to obtain similar results (Chong et al., 2008).

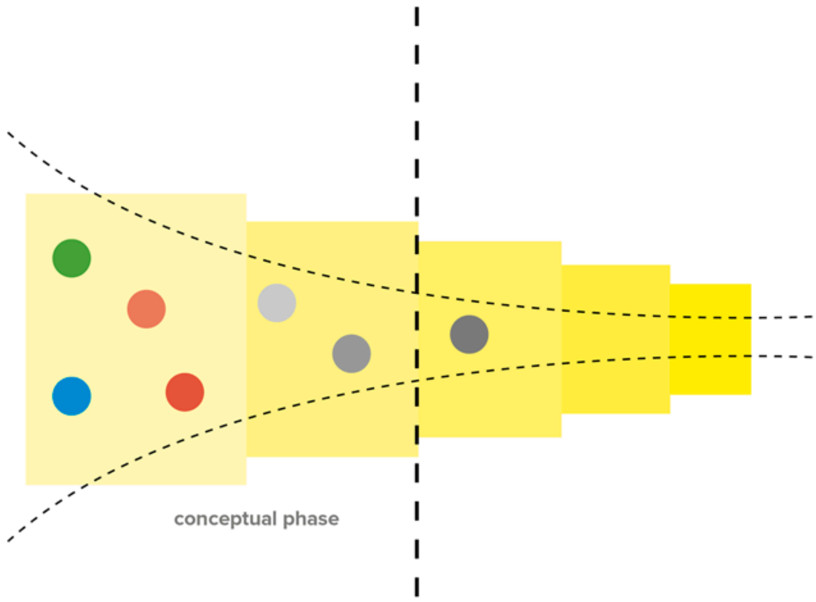


Fig. 13 – The design process.

The software construct presented in this thesis will assist the designer at the very beginning of the design process. The model is a loose construction of software applications and property scripts which are united with a common goal and a common objective. The software construct is divided in four parts (Fig. 14) - or four phases - which are also part of four nesting iterative processes, which will be explained in more detail in the following paragraphs.

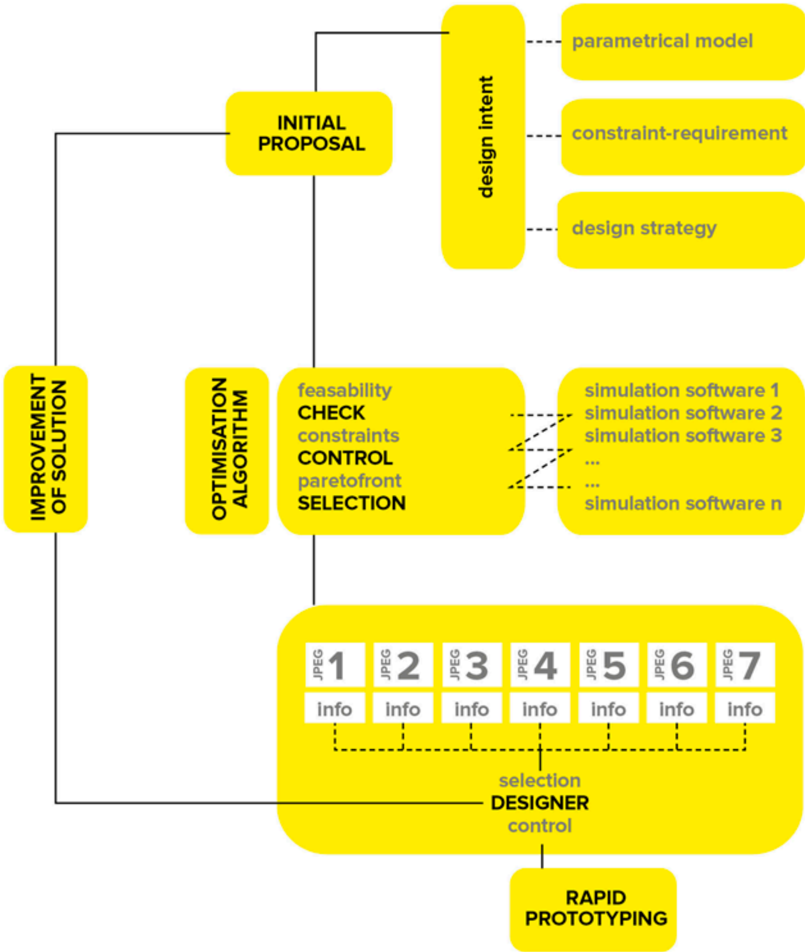


Fig. 14 – General Outline of the Software Construct.

3.2.3 Preparing for Fuzzy Front End optimising

In order to start a design process using optimisation, three initial conditions have to be fulfilled in order to define "Design Intent". The designer has to define a design strategy in accordance to the kind of design project which will be developed. Only certain characteristics of the initial parametric geometry will be used and can be used in the process of optimisation. How a design object can be optimised and which performance parameters will be used as optimisation criteria are, of course, paramount to the success of the optimisation model. It is therefore necessary to explore different possibilities right in the beginning of the process. This iterative process is absolutely necessary for testing the integration of multiple constraints into a functional design explorer (Hernandez, 2006).

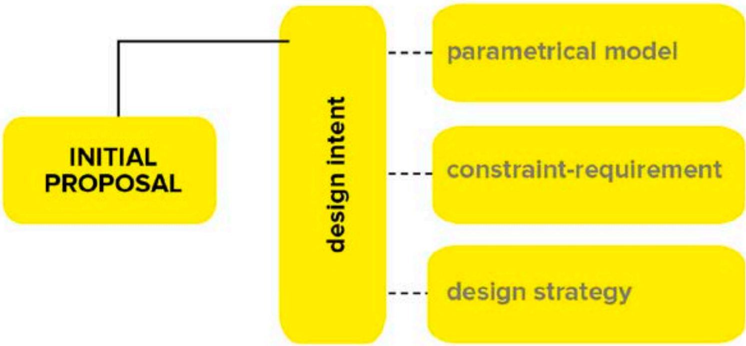


Fig. 15 – Preparing the software construct and the parametric geometry.

The proposed model will act on certain predefined characteristics of an initial parametric geometry, and, it will do so in relation to a strict set of rules, boundary conditions, performance requirements and structural constraints which were formulated during the information phase of the design project. Some of this information has to be introduced in the specific simulation software program, while other information will be essential to the functioning of the scripts written specifically for this software construct. This initial parametric geometry will than act as the input of the system and will be prepared for optimisation (Fig. 15). The parametric geometry has to be translated to the appropriate file format for importation in each specific simulation software program. The CAD tool used to produce the geometry for the case studies in this thesis, is Rhino3D due to its versatility, but any CAD tool can be used, which is able to produce the information in the necessary file format that can be analysed in the simulation software application.

3.2.4 Parametric Geometry

The advent of sophisticated digital modelling systems has enabled designers and architects to create, manipulate and control very complex geometries and forms. Unfortunately most of the CAD and CAAD software program packages used by designers and architects are free-standing, closed products, developed with a particular set of goals in mind and with output in proprietary file formats. Communication between two different software programs can be very difficult and limited. The strategy of linking different software applications and how and which format to exchange information between the different platforms must be well reflected upon and prepared by the designer.

Many different modelling techniques can be used as a base for the FFO. Most common and most popular is parametric feature based modelling. The particularity of this kind of modelling allows for parametric variation, where changes to a designed geometry do not alter the basic characteristics of this geometry. The concept of parametric variation is accomplished by making the model constraint based and dimensionally driven. The most common approach used by designers in generating shapes is the direct use and manipulation of software tools such as points, lines, lofts, sweeps, etc... Those kinds of tools can be found in the commercially available CAD software environments such as Rhino 3D, Catia, SolidWorks or Microstation.

As a methodology for form finding, anything between the simple transformation of basic forms and the creation of form by manipulation of code has been used by designers in the process of form generation. Even real time interaction has been incorporated in the definition of a geometry allowing for the changing of shape according to external functions (Oosterhuis, Xia, & Hyperbody, 2009). A set of finite instructions which takes parameters as inputs has been used to generate parametric models where geometrical components are considered as variables (Hernandez, 2006). More recently, there has been growing interest in using external factors as the driving force in the generation of form. Morpho-Ecology is an example of such an approach based on a framework for architectural design rooted within a biological paradigm (Hensel et al., 2006) (Hensel & Menges, 2008). Some software environments such as ParaCloud Modeler, Genr8 (O'Reilly and Hemberg, 2007) (Hemberg, 2011) and Bentley's Generative Components have been developed especially for this kind of design approach.

An alternative approach to the use of parametric variation in form geometry is the use of shape grammars (Knight, 2004). Especially in the field of architectural design, the theory of shape grammars has been used as a framework for the study of layout problems or for the application of topological modification of the original form in a process of form finding. Shape grammars are a set of rules which apply to the arrangement of shapes in space. Those shapes can be either two dimensional geometric figures or three dimensional and additional labels can be defined such as colour and material. The rules are based on Boolean operations such as union, difference and intersection, and transformation operations such as rotation, reflection, scale or any combination between them. The application of shape grammars in the field of architecture has been pioneered by Stiny and Gips (1972). Different techniques and tools have been successfully applied in specific design problems, for example, in revealing the structure of the medina of Marrakech (Duarte, Rocha, & Soares, 2007), in the exploration of different configurations based on the style and layout of Palladium Villas (Stiny & Mitchell, 1978), or in three dimensions with rapid prototyping (Sass, 2007) or the Prairie style houses of Frank Loyd Wright (Koning & Eisenberg, 1981). Topological optimisation with shape grammars has been used by Shea (Shea et al., 2005), but was limited to the exploration of different truss structures (see also 2.3.3). However, the application of shape grammars and the topological optimisation of building envelopes have been appointed as an important future area of research (Kilian, 2006).

3.2.5 Optimisation Algorithm

Which kind of optimisation algorithm is best to use in each specific situation and each kind of optimisation problem is open to discussion (Renner & Ekárt, 2003). Many different methods have been proposed and tested, some have been used for optimisation in the field of architectural design, but most of the techniques for optimisation have been developed for applications in other fields, for example in the field of engineering design. Simulated annealing (Shea et al., 2005), Gaussian adaption (Hinterding, 1996), hill climbing (Carvalho, Lavareda, Lameiro, & Paulino, 2011) etc., are all different and possible options for integration in a model for interactive optimisation, and some of the techniques were effectively applied for optimisation in the field of architectural design. Recent renewed interest in optimisation strategies and their application on practical problems, such as for example research into a multi-objective optimisation model applying Tchebycheff programming for building retrofit strategies (Asadi, da Silva, Antunes, & Dias, 2012) or Genetic Algorithms have been applied to the study of ergonomic chair design (Brintrup, Ramsden & Takagi 2008).

For this study a Multi-objective Genetic Optimisation Algorithm will be used. It is based on a heuristic general optimisation algorithm slightly modified to fit this research and to meet logistic limitations (Fig.16). Originally the algorithm, a Reduced Pareto Set Genetic Algorithm with elitism (RPSGAe) has been developed specifically for the optimisation of single screw extruders (Gaspar-Cunha, 2009) and has been successfully applied and tested for multi-objective optimisation problems with big populations (Gaspar-Cunha & Covas, 2004).

Genetic Algorithms have been criticised in general (Abel, 2007), but also more specific also because of the tendency to converge towards local optima within a neighbouring set of solutions, sometimes even converging to a single arbitrary point, rather than searching for a global optimum solution from within all possible solutions. There seems to be no general solution to this problem because all proposed alternatives have their particular drawbacks. Niching strategies, as used in NPGA's (Niched Pareto Genetic Algorithm), and clustering strategies are the most common techniques to circumvent this problem and maintain the needed diversity among the individuals. The software construct discussed in this study uses a Reduced Pareto Set Genetic Algorithm with Elitism (RPSGAe), which is a modified EA which proved to successfully avoid the problem of deterioration of fitness of the populations during the successive generations. It has been benchmarked and compared to a NPGA's, and shows equal performance with the added value of reducing the final Pareto set which is a clear advantage when the optimisation problem requires the use of large populations (Gaspar-Cunha, 2009).

Evolutionary Algorithms were inspired or modelled on the principles of natural evolution and obviously nature has abundance of resources and above all time. For man-made systems however, computational complexity is prohibitive, even in problems which are not even that large and complex. But, in nature, the survival of the fittest is not about exact measures, it is rather a ranking among competing peers. And it is this natural tolerance for imprecision that will be explored, but instead of mimicking this process by applying complicated techniques such as Fuzzy Fitness Granulation (Davarynejad, 2008) and by introducing selective fitness computing (Torres & Sakamoto, 2007), the selection and ranking among individuals will be done by the designer.

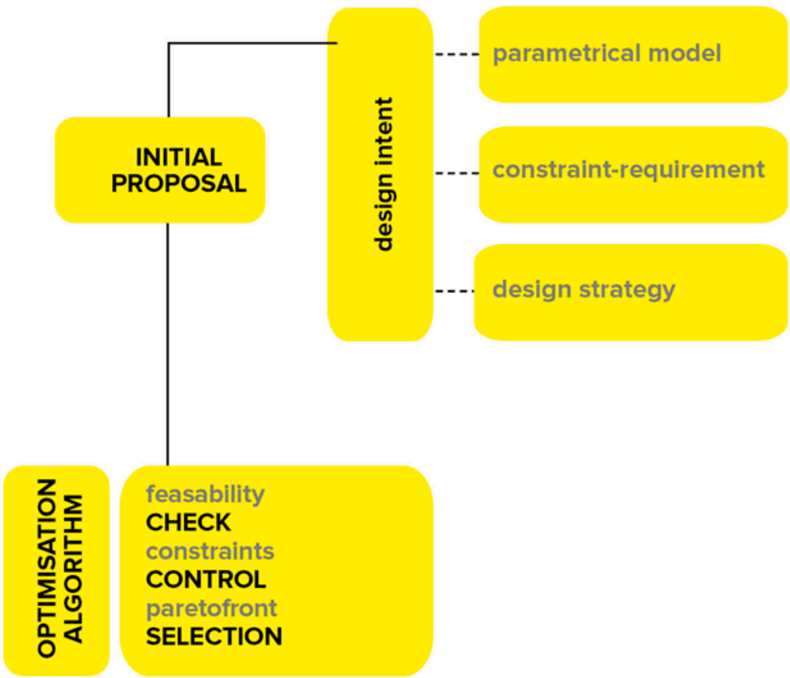


Fig. 16 – Coupling of the Optimisation Algorithm.

A practical algorithm for evolutionary computing adapted to the purpose of selecting valid design ideas and proposals needs to be robust, flexible and easily adaptable to the specifications of the optimisation. Although it is possible to develop customised algorithms to fit in with this software construct, it is not the task of the designer nor does he generally have the necessary mathematical skills to develop and adjust such an algorithm. The strategic selection for optimisation and the development of precise algorithms is a highly specialised and very particular part of the field of soft computing (Brintrup, Takagi, Tiwari, & Ramsden, 2006).

A fundamental step of optimisation strategies using EA's is the definition of its parameters such as crossover and mutation rates and population lengths. Choosing the appropriate parameters does increase the quality of the final results and reduces the time needed by the computations. By using the RPSGAe developed by Gaspar-Cunha (2004) it is possible to build upon the results of a large series of studies previously done by this author, and the parameters in those studies can be used as starting point for the optimisation algorithm used in this thesis, providing a valuable contribution to the robustness of the software construct (Fig. 17).

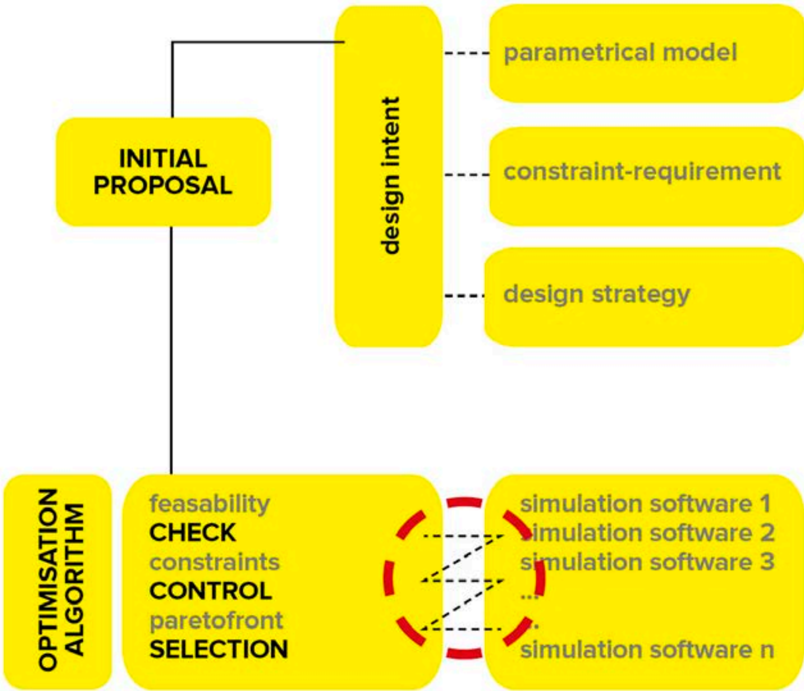


Fig. 17 – Iteration between EA and analysis and simulation software.

3.2.6 Analysis and Simulation Software programs

In the last 50 years, hundreds of building energy efficiency simulation programs have been researched and developed (Crawley, Hand, Kummert, & Griffith, 2008)[1]. Many of the existing software programs are very specific in use and most were also developed within research departments of universities, and thus they are not commercially available. Possible useful software programs on integration of a software construct built for optimisation; have to be evaluated mainly on their power to exchange information in a useful format between the different parts of the software construct.

The most common evaluations applied on any designed object are focussed on the structural and dimensional characteristics of that object. The simulation tools applied for this evaluation are programmed in a way to give specific answers to known questions. Qualitative results will give advice to the designer for further development; quantitative answers can serve for further optimisation of the objects performance. In both cases, however, the results which are obtained are always predefined by the kind of simulation tool implemented, and are usually an afterthought and not an integrated part of the design process (Schwede, 2006).

In simulation software applications the aspects of a design proposal to be considered are mathematically modelled. However, limited knowledge of all interactions within a system usually prevents one from producing a simulation environment which replicates the original in every detail. It is therefore important that the right simulation software program is selected in function of the parameters and the constraints which are most important for design intent. Evaluation criteria are as much part of the design process as the final results.

Digital simulation has some significant drawbacks, however. The software program cannot be applied without profound knowledge of the field. Otherwise it is not always clear which parameters or constraints are directly involved in the results of the simulation, or how they influence the results. Simulation results may also give no indication about what has to be altered or

[1] See also the Building Energy Software Tools Directory provided by the U.S. Department of Energy at http://apps1.eere.energy.gov/buildings/tools_directory/subjects_sub.cfm, accessed 10-01-2012.

how it has to be changed. Furthermore, the simulation software program has to be capable to produce reliable results for that specific configuration which has to be optimised, and it has to produce the results in a format which can be interpreted by other software applications, more precisely by the EA which will be deployed in this model. It is after all this iterative process between generation and simulation which is the motor of the proposed optimisation model.

3.2.7 Interaction with the designer

The designer is not a passive observer of the optimisation process. In the present software construct, after each generation, a selection of solutions from the Pareto Optimum results is presented to the designer as an image with a summary of the simulation results. By showing a pallet of solutions instead of only one best solution the common problem of "design fixation" can be avoided (Jansson & Smith, 1991).

Interactivity between the designer and the software construct is then provided through the selection of one or more desired or aesthetically interesting proposals amongst the Pareto Optimum solutions (Fig. 18). These selected individuals can either be used for breeding or for mutations in a new iterative cycle of optimisation. If only variations of one selected individual is desired, both the parent and child populations will be populated with variations of that one chosen result. It will also be likely that sometimes none of the generated results are corresponding to proper or desired aesthetical expectations or that the quantitative results of the Pareto optimum solutions are just too low to be acceptable. In this case it would be necessary and desirable to stop the optimisation process in this iteration cycle and restart the whole process with a new or a changed parametric model.

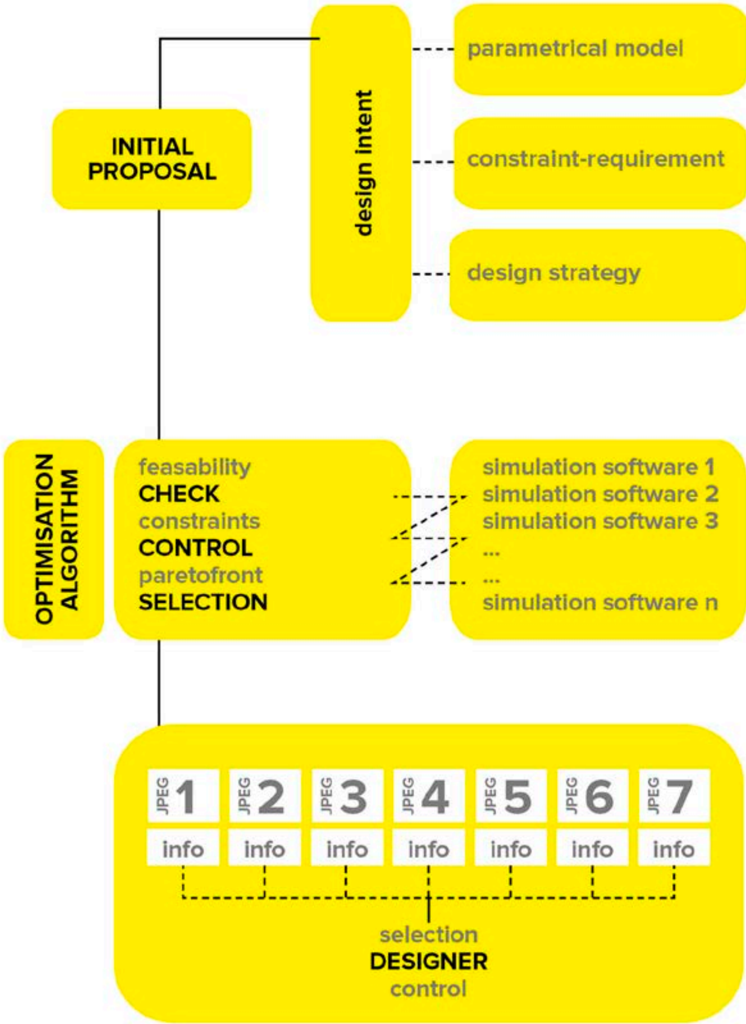


Fig. 18 – Optimisation -Construct.

In the current construct, the intermediate results of each optimisation cycle are archived in a separate directory, and thus any old solution can be recalled from this file any time and used for introduction in the present running optimisation cycle. It is expected that this unorthodox use of optimisation can and will result in more creative exploration of the solution space.

3.2.8 Iteration cycles within the software construct

Providing the means and conditions for massive iteration is the most important condition for successful optimisation. As was explained in paragraph 3.2.7, the existence of multiple iteration cycles is a characteristic of the conceptual phase of the design process (Fig. 19).

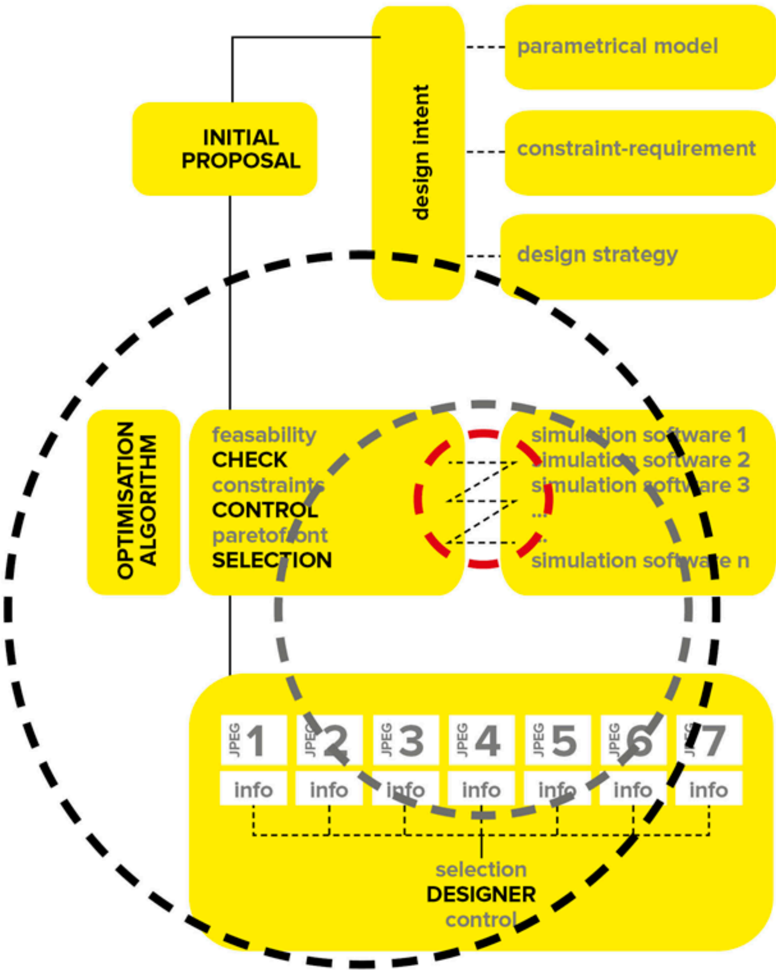


Fig. 19 – Different iterative cycles within the software construct.

Once the designer has found a suitable solution or once he has gained enough insight and understanding of "good" or "best" solutions, it might be appropriate to explore this solution or this line of solutions in more detail. This might be considered the end of the conceptual phase of a design process and the beginning of the detailing phase of a design process.

3.2.9 Rapid Prototyping

In the last two decades, the advent of accessible rapid prototyping has reinstated the importance of the fabrication of physical models as an important part of the design process, or even as the central research activity of a digital design method. Architects such as Sass are developing a two-stage method that integrates generative computing with rapid prototyping which supports physical evaluation of the designed object (Sass & Oxman, 2006).

The introduction of CAD technologies enabled designers to develop true free form building envelopes based on complex surface geometries. The availability of digital manufacturing processes and equipment allowed these forms to be tested on economic and technical feasibility. This allowed for research in the practical application of integration of generative systems in the design process on many Universities and Architecture Schools, Faculties or Departments. A research project on the design and digital fabrication of low cost housing for the developing world (Sass & Botha, 2006) of the MIT Digital Design Fabrication group is an example of such a practical application. Their project proposes an automated generative method using shape grammars, to design customised houses, based on predefined parameters with a set of previously assigned variations, in a process of shape selection. The chosen final proposal is laser cut out of plywood on a 1/10th scale. This scale model permits the confirmation of the construction viability, allows testing the connections between the different parts and enables also for a subjective evaluation.

Physical model making has always been part of the design process, and has been used with two different objectives. Small, coarse models are used to realise mental concepts, for testing or for the validation of perceptual evaluation such as proportions. It is in this way that the construction of physical models is used as a design method. Before the advent of the computer and the use of powerful 3D modelling software programs for visualisation, or virtual prototyping, more or less elaborated scale models were a preferred method in any area of design. The uses of these models and the skills to build them have declined over the last decades and are almost completely banned from the contemporary design processes, where almost perfect photorealistic virtual models can be built in the same time span and with less effort, even in the conceptual phase of the design process. However, technological progress of digital manufacturing techniques and the availability of those techniques have reinstated the importance of functional models and prototypes during every phase of the design process. Digital design fabrication can thus be strategically integrated in the design process and the physical results are excellent means for evaluation of any kind of parameter.

The other important reason to build physical models was purely a mean of clear communication of design intent or a presentation of the final result. Before the generalised use of drawings and blueprints, scale models with great detail were exclusively used to communicate the appearance and the working of a building, a machine or an object to the people who were supposed to build a functional prototype. The presentation of a detailed physical scale model was also important whenever the continuity of the design project depended on a non-specialised decision maker. The owner of the project is not always instructed in any particular design field, and does make decisions based on trust and emotive response.

It is possible to develop a design project completely in a digital computer environment. Obviously there are many other ways ideas can be generated and tested. Physical models were a preferred and still are an approved technique for conceptualisation of design ideas. Many complex buildings, such as churches and cathedrals built during the medieval eras are testimony of the successful use of these techniques. Even contemporary successful architects like Frank Gehry still use models as a basis for generating and exploring ideas (Foster, 2003). In many of his projects he starts with a paper model developed by hand which is subsequently scanned. The data is then used to build a computational model for further editing and detailing. Sometimes this digital model is again converted to a physical model for further manipulation by hand. This iterative process proceeds until the design intent is satisfied.

Currently, powerful analysing software programs are readily available for any architect or designer. Software applications with a short learning curve and with an attractive and user friendly graphical interface allow for the computational simulation of the design objects' performance before it is built and available for investigation in the physical world. Performance issues differ from one design project to another, and the testing of a model in the physical world or under controlled laboratory conditions cannot or should not be ruled out of the contemporary design process. Another reason to implement the production of scale models or prototypes during the conceptual stage of the design process is the fact that some of the qualitative characteristics of an object are highly subjective and the object of personal interpretation. Sometimes designers have no adequate way to evaluate significant aspects of their design without a material example. It is the author´s conviction that model-making incites on open-ended design projects, paramount to innovative solutions where exploration is the objective for probing for possible solutions.

3.2.10 Graphical User Interface

The results of the iterative optimisation cycles within the software construct are collected in a database or on an Excel data sheet. However they are but numbers. Evaluation by the designer and decision making among alternative design options requires simultaneous side-by-side presentation for visual comparison of possibilities. By using a graphical user interface (GUI) the designer can maintain control over the process and the process will be quicker, more satisfying and probably with better results. Therefore the model should use two kinds of interfaces. To build the parametric model the designer can use the regular GUI from the CAD software. This interface should be familiar for the designer who does need a minimum of operative expertise to be able to build a suitable parametric model. The second interface is a Graphical User Interface developed on purpose for the FFO Method and which is the interface used to interact with the overall software construct during the design process.

A Graphical User Interface is desirable as the necessity and possibility of taking rational decisions during the design process is very much based on visual representation of intermediate design solutions. This is a traditional and very important way of designing for an architect or a designer. In order for this software construct to function within a designerly way of working and thinking, a GUI was developed in VB2008 Express. This software package can be freely downloaded from the Microsoft servers. It is well-documented, easily scriptable and not too difficult to master.

4 Assessment and Testing

4.1 *Case Study 1*

In order to implement and test the effectiveness and suitability of the construct, several case studies have been made, each one combining and integrating an increasing number of complexities in the problem to optimise. The case study described in this chapter is the result of this process of exploration and tweaking of different software's and different components of computer programs. The main objective of this case study is the development and implementation of a fully functional software construct with the focus on integration of the CAD software, the simulation software and the optimisation algorithms. As a result this functional software construct will guide the optimisation process and enable interactive behaviour between the (human) designer and the process with the objective to produce preferred and better solutions. A second objective of this case study is to explore ways of communication between all three of the different parts of the software construct proposing a smooth and seamless transfer of information to guarantee the robustness of the possible solution candidates.

4.1.1 The Experimental Set-Up

The software construct is intended and modelled as a method, as a systematic and planned process or procedure based on the correct use of design skills and techniques. Considerable effort is made to avoid building a software *tool*. Such a tool would force the designer to act according to strict and preconceived intentions (see also 3.1.1) and the output which could be produced would be limited and highly expected. What the output would be, or better, could be, depends however largely on; how the digital model is constructed in the first place, what is the design intent behind this project, how is the model allowed or planned to change and between which boundary conditions this can happen. How those transformations are possible, how those transformations are able to result in interesting solutions, or how this model can be seamlessly integrated in the overall software construct, does not only depend on the software environment where the parametric model was created but also in part on the knowledge and the operational and technical expertise of the designer.

To test and tweak the software construct, a simple geometry was used with no specific function, but with the possibilities to illustrate clearly and visually how changes on the parametric variables affect the final three-dimensional outcomes. This simple geometry could represent a basic "roof-like" structure made out of a lightweight composite material. With this set-up it will be possible to calculate the area of the "dome", which gives feedback about the lightness of the construction. This can be one of the design objectives of this first case study and one of the goals to optimise to: less surface area means less material and thus a lighter structure. If this "dome" would be built using a composite material or any other material for that matter, the structural feasibility and robustness can be simulated and calculated. On the other hand, a "roof" can also be related to "area covering", and thus to shading, which can justify the simulation and analysis of lighting and make it visually understandable. Lighting (or in this case shading) can therefore be the second goal for optimisation. Less daylight under the structure the more effective it would be as a shading device and the more it would correspond to the initial design intent: this way a design explorer has been constructed (see also 3.1.4), apt for design exploration. The building of the parametric CAD model together with the defining of the boundary conditions and the limits of variation of the control points are the first act of design in a digital design process.

4.1.2 Modelling the Initial Parametric Proposal

As a starting point for this research project, a generic dome-like geometry was built in Rhino 3D CAD software. This 'roof structure' has a dimension of 5m x 5m, represented by a single Non-Uniform Rational Basis Spline (NURBS) surface. Using only a single surface makes the exact mathematical representation of a free form surface possible and allows for precise control by manipulating the control points of the NURBS surface. In a NURBS surface the 'control points' determine the shape and curvature of the surface and a single control point only influences those intervals where it is active. Using this method some parts of the surface can be changed while others are kept equal. The manipulation of the control points allows for a kind of parameterisation of the surface. Control points in this case are used in the everyday meaning of the word 'point', a location in 3D space defined by its three coordinates in the X, Y and Z planes.

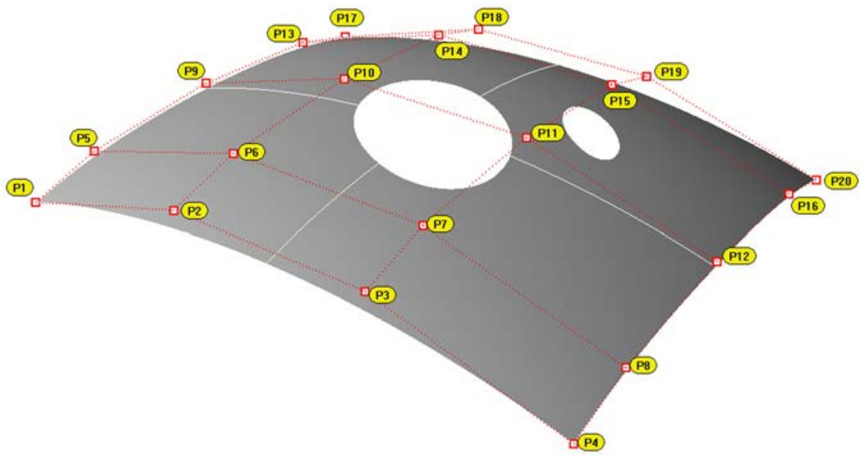


Fig. 20 – "Roof structure" with the NURBS geometry defined by 20 control points.

In the present study a set of 20 control points were defined allowing for virtually unlimited adaption and deformation of the surface geometry based on the coordinates of the 20 control points. Any similar NURBS surface can be specified by either less or more control points what allows for a rougher or a more subtile manipulation of the final outcome. The quantity of control points is directly related to the amount of variables which have to be manipulated by the optimisation algorithm and has thus a direct influence on the computational time necessary to calculate every instance in the optimisation process. As a compromise between geometric variation and available computational power a grid of 20 control points was decided upon, which resulted in 60 variables.

In the generic 'roof structure' two holes were rather randomly designed without any dimension or fixed place (Fig. 20). In this way, visual feedback could be obtained as to provide an indication of the correct functioning of the simulation software: numeric average values alone do not give an adequate understanding of eventual localised discrepancies, hence the use of this simple feedback construction. Furthermore, the holes also contribute to the aesthetic perception of the object. Strangely twisted holes are probably not experienced as aesthetically pleasing.

Before the first optimisation run, extended manual modifications of the control points were tested and the results allowed for the clarification and the setting of limits to the variation of the coordinates of the control points, information which was necessary to limit the solution space to reasonable and feasible results for the free form surfaces (Fig. 21).

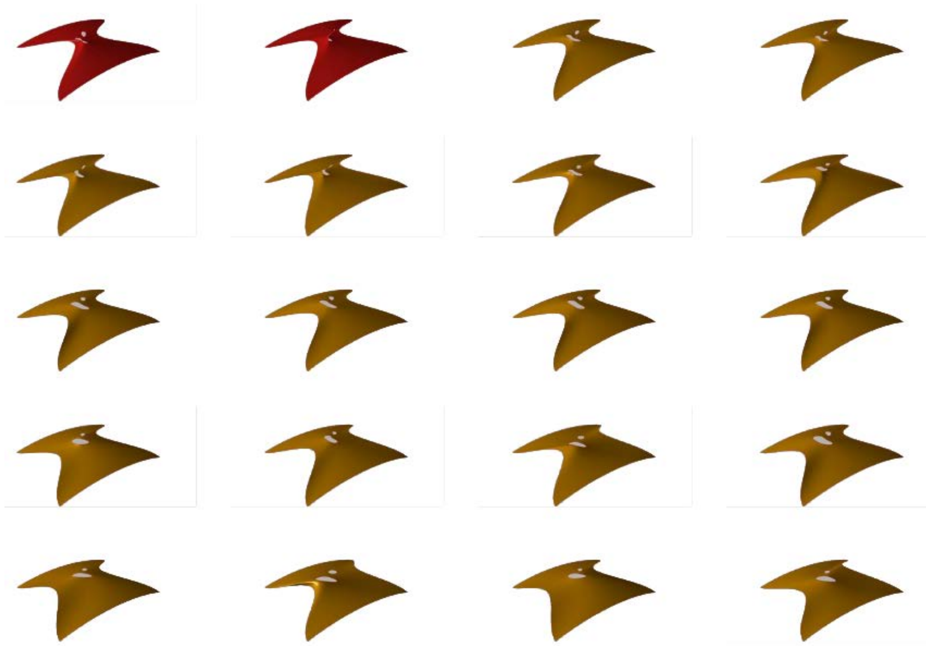


Fig. 21 – Testing of the variation of the control points – similar results.

However, this process has to be executed with extreme care. If the range of variation is allowed to be large many results will be produced which for sure will contribute to a huge variation in aesthetical possibilities, some completely unexpected, but maybe not suitable as a "roof like" structure (Fig. 22).

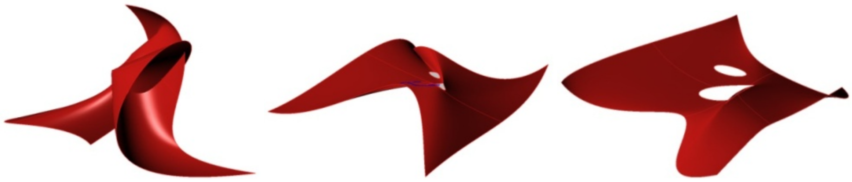


Fig. 22 – Testing of the variation of the control points – unexpected results.

If on the other hand the range of variation is too restricted, large areas of the (possible) solution space will be unexplored and desired interesting solutions can be missed (Fig. 23).

Fig. 23 – Testing of the variation of the control points – too little variation.

Therefore careful selection of constraints and limits are of crucial importance in this optimisation process (Fig. 24).

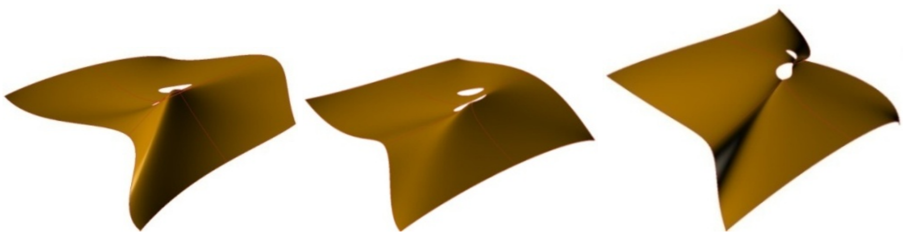


Fig. 24 – Testing of the variation of the control points – with folds.

In the present construct, the manipulation of the control points will change the shape of the surface, after which the CAD software will calculate the area - as an indication of the weight of the structure - and export the surface to a building analysis software for subsequent numerical analysis, in this case the average Daylight Factor under the structure, as an indication of the light functionality of the structure. The results (Area and Daylight Factor) are saved for subsequent use by the optimisation routine.

The examples presented here make it clear that optimisation without proper preparation of the parametric geometry does not allow for optimisation. This is the first iterative sequence of exploration as explained in detail in paragraph 3.2.3. Iteration in this phase of the design process will help the designer to resolve two important questions at the start of a digital design process: does the geometry allows for significant modifications and can one expect solutions within acceptable aesthetical boundaries? The perception of technical feasibility might or could also be considered as a part of the evaluation process of the designer, but technical or constructive viability is usually an important yes-no parameter in any process of optimisation. An object which is not viable for construction or fabrication cannot (or should not) be considered apt for optimisation in the first place. However, in the conceptual phase of the design process, unrestricted exploration might contribute to better understanding of the design problem.

Opting, in this case study, for a simple NURBS surface with a limited number of control points, allows for detailed exploration of the exchange of information between the different components of the software construct, demonstrating the technical feasibility of this optimisation method. The use of a GUI, as proposed and described in paragraph 3.3.10, was not considered because of the necessity to monitor the behaviour of the different software components in close detail.

4.1.3 Multi-Objective Optimisation

The problem to be solved in this case study has three objectives to attain. The first objective used for optimisation is the minimisation of the area of the ‘roof structure’, which is a measure of the effective use of material and the ‘lightness’ of the structure and is assumed to result in a more efficient solution. This area can be calculated automatically by the 3D CAD software and the results stored. The second objective for optimisation is the minimisation of the average Daylight Factor under the structure. This relates to the objective of eventually constructing such a ‘roof structure’: a smaller Daylight Factor should be a measure for more coverage, and more coverage is the main goal of such kind of a structure. The third objective is the Designer’s personal perception and interpretation.

For this kind of optimisation problems, Multi Objective Evolutionary Algorithms (MOEA) are a tested and proven method (Gaspar-Cunha & Covas, 2004), and the generation of a Pareto frontier after the optimisation is an objective way of visualising of the trade-offs between both quantitative objectives, as clearly demonstrated in the research work of Caldas (2006). How to refine the formation of a Pareto frontier and how to ad information on the relative importance among the objectives is the object of a present ongoing research by Ciftcioglu and Bittermann (2009). The goal of this research is clearly focused on a more user-friendly application of the Pareto frontier for use as routine, practical and non-academic optimisation.

Since the third objective proposed for this study is not quantifiable but, additionally, is very subjective and based on personal visual interpretation, an iterative strategy of optimisation has to be applied, which takes into account the preferences of the designer (in soft computing usually referred to as the decision maker or DM). This iterative process is the core of the optimisation method and consists of a process where the optimisation algorithm (MOEA) generates a set of optimised solutions, a Pareto front, based on the selected objectives, in this case Daylighting and surface Area. Then the decision maker selects the preferred solution from within an area of this Pareto frontier (Fig. 25), according to his or her aesthetical perception, eventually related or biased also to a specific quantitative objective. This information is reintroduced in the MOEA for further optimisation. Such a process can be repeated until eventually a satisfactory solution is found by the designer, or until the necessity of restarting the whole process is considered if no solution can be obtained which is satisfying, as has been exemplified in paragraph 3.2.8.

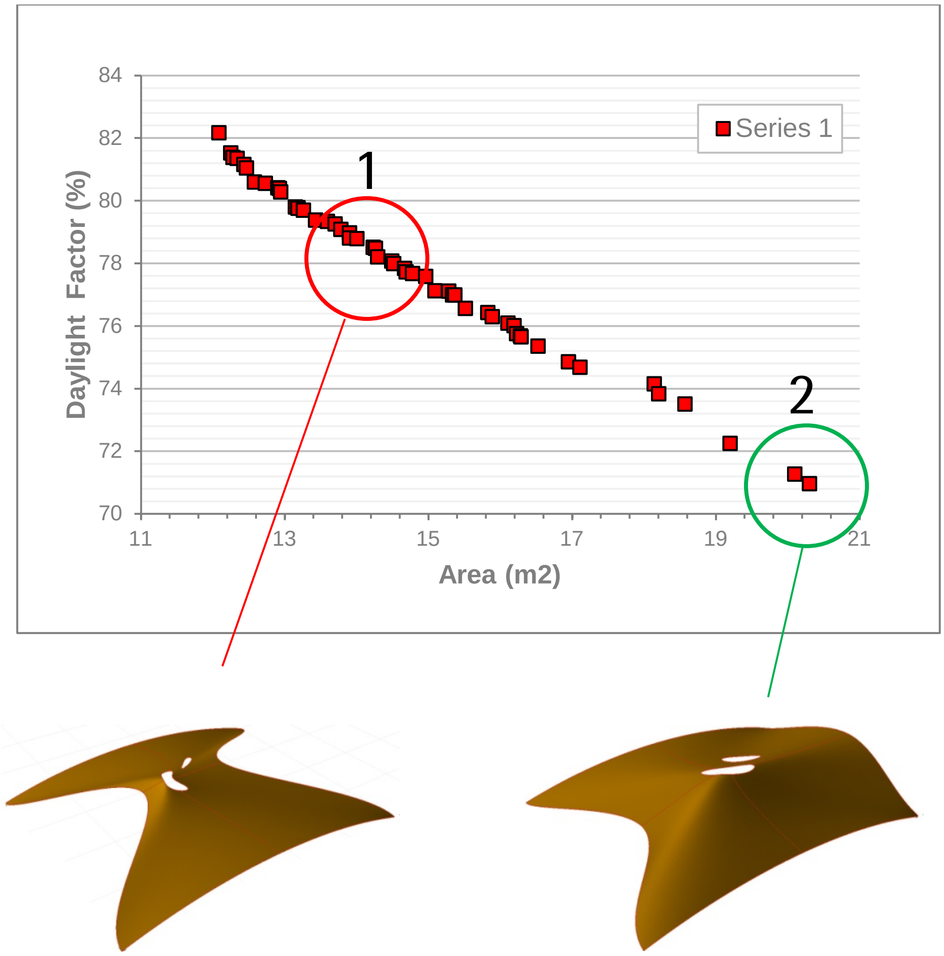


Fig. 25 – Different regions represent different aesthetical solutions and trade-offs between Daylight and Area.

The solution for this kind of optimisation processes involves the articulation of the preferences of the designer. This is exemplified in figure 25, different solutions can be selected from different regions on the Pareto frontier. All the solutions on a Pareto frontier are optimised solutions, albeit with a different measure of relative importance of the objectives in consideration. Both objectives for optimisation in this example have to be minimised, with the solutions concentrated in region 1 having a worse performance for the objective *Daylight Factor*, and the solutions in region 2 having a worse value for *Area*. A balance between the performance relative towards the two objectives, can be found in the solutions in the centre-region of the Pareto frontier.

Different methods or strategies can be chosen to introduce multi-objective preferences in the optimisation process, depending on how the process of search is connected and combined with the process of decision making. The most common and most easy way to deal with multi-objective optimisation in design problems is by articulating the decision maker's preferences *a priori*. The decision maker defines the variables for optimisation and the range of variation and the goals for optimisation before running the MOEA. This means that before the actual optimisation a factor of relative importance is attributed to the different objectives for optimisation (Anderson, 2001). This strategy however implies a profound knowledge about the limits and the interplay of those objectives, because specifications should be goal oriented and objective.

Another optimisation strategy will first search the solution space for a set of Pareto Optimum solutions and present them to the decision maker. According to specific and case-dependent process knowledge, an iterative process of trial and error with new information provided by the decision maker is added before each optimisation cycle. Just as in the first strategy the decision maker does have to define the optimisation variables and their range of variation: however, what is different in this case is the tweaking of results, eventually in function of the aesthetic outcome.

One of the goals of this case study is to test the software construct with the intent to guide the optimisation process and to introduce some kind of interactivity in the process with the objective to produce a preferred *and* an optimised solution. Therefore optimisation and selection (decision making) are done at different steps of the process. At each step preferred intermediate solutions are chosen or selected by the Designer (decision maker) and this information is used by the Multi Objective Evolutionary Algorithm (MOEA) as input for generating better alternatives, all this in an iterative process. The mathematical methodology used in this process depends on the kind of MOEA selected for optimisation and the different methods applied to incorporate the references. A detailed explanation of the mathematical theory underlying this process does not contribute to the arguments in this thesis. The MOEA used in this case study has been proven in previous research to provide results in line with the objectives of this study, details of which can be found in e.g. Gaspar-Cunha and Covas (2004). Within the scope of the current project the author relied on the close collaboration and the specific expertise of the developer of this particular MOAE for the success of this first case study.

4.1.4 Optimisation Strategy

In this Case Study a combination of the two different optimisation methods or strategies described before have been used, and applied in two different phases of the optimisation process.

The combination of those two methods is illustrated in Fig. 26. Optimisation starts by the definition of three different optimisation runs, each one characterised by a different set of limits, constraints and range of variation imposed by the Decision Maker (designer or architect) on the decision variables. Subsequently, each one of these cases is optimised individually and independently. Out of the pool of combined results of all these optimisation routines, a new set of solutions will be selected and used as the initial population of a last optimisation process. This last optimisation process could be a simple optimisation routine, similar to the previous ones, or it could be an iterative process as described in the preceding paragraph. It can be expected that the final results will have characteristics of all the previously selected preferred solutions.

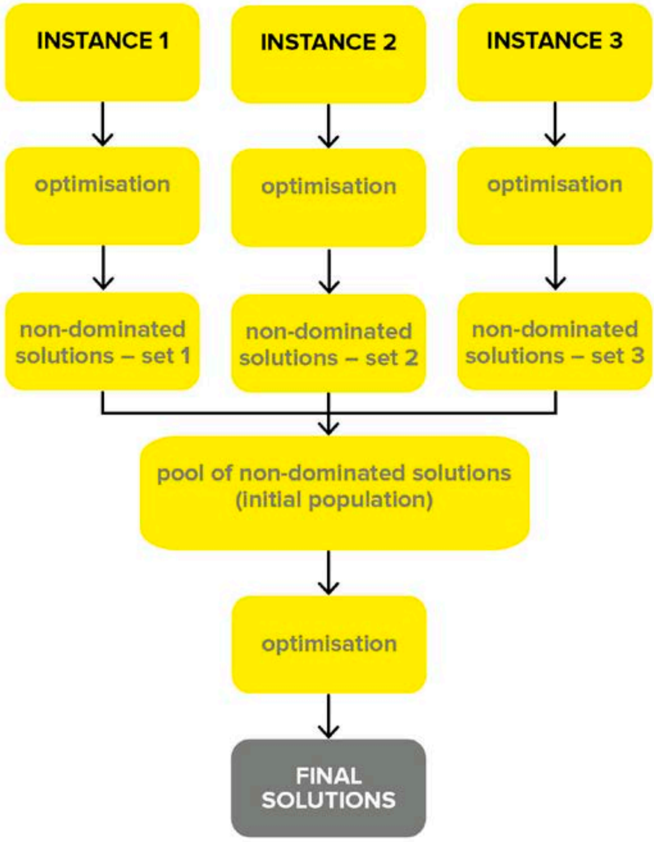


Fig. 26 – Structure of the optimisation strategy for case study 1.

4.1.5 Testing Interactive Optimisation

In order to explore different conceptual solutions, three different geometrical boundary conditions were used, each one leading to a different subset of solutions, according to the method described and illustrated in Fig. 26.

In the first case, the less controlled one, none of the control points of the NURBS surface were restricted and the coordinates were allowed to randomly vary in the range of 0.5 and 5 meters. But, even after 10 optimisation cycles, no interesting or feasible results were produced. Mostly self-intersecting surfaces were generated, and although this kind of surface cannot be used for structural analysis and thus as a start for a detailing design process, gently twisted surfaces can still be recycled as input for a more restricted optimisation. However, and although a Pareto Optimum frontier has been created and all those solutions belong to a population of best results, none of them can be used directly for generating a feasible roof-like structure (Fig. 27). It is clear from those results that different boundary conditions have to be introduced in the optimisation strategy and that this procedure is absolutely necessary and most important for producing more "useful" results.

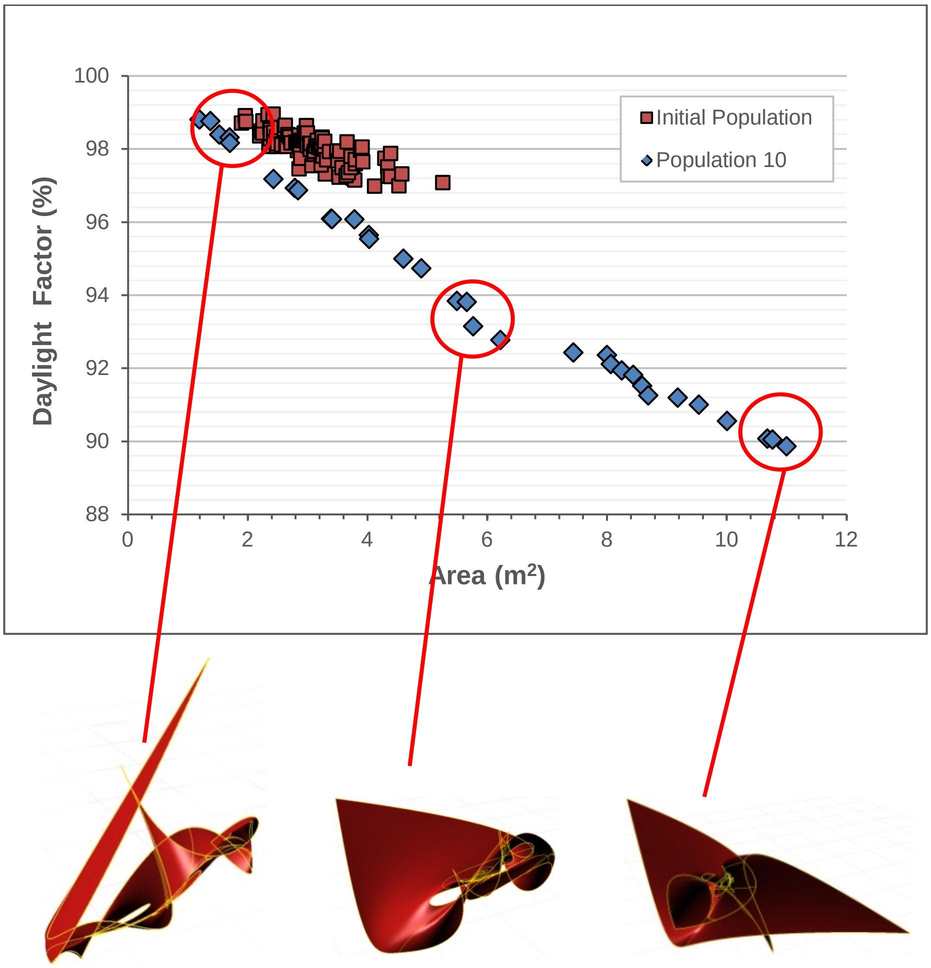


Fig. 27 – Optimisation run with no restriction to the control points: Pareto Optimum with some results.

In the second case the corner points of the 'roof structure' were fixed and not allowed to vary (no change allowed to any of the three coordinates of those points). This procedure thus limited the set of decision variable to 48. As in the previous case, the coordinates of all other points could change in the range between 0.5 and 5 meters. Fig. 28 shows the results after 10 optimisation cycles.

It is clear from the results that fixing the corner points alone, does not generate a Pareto Optimum set of solutions with a reasonable possibility to serve as a starting point for more detailed designing. However, some of the results may indicate interesting directions and arouse more specific attention for an eventual restarting of the optimisation process within a different iterative design cycle. But again further restriction of the boundary conditions has to be considered.

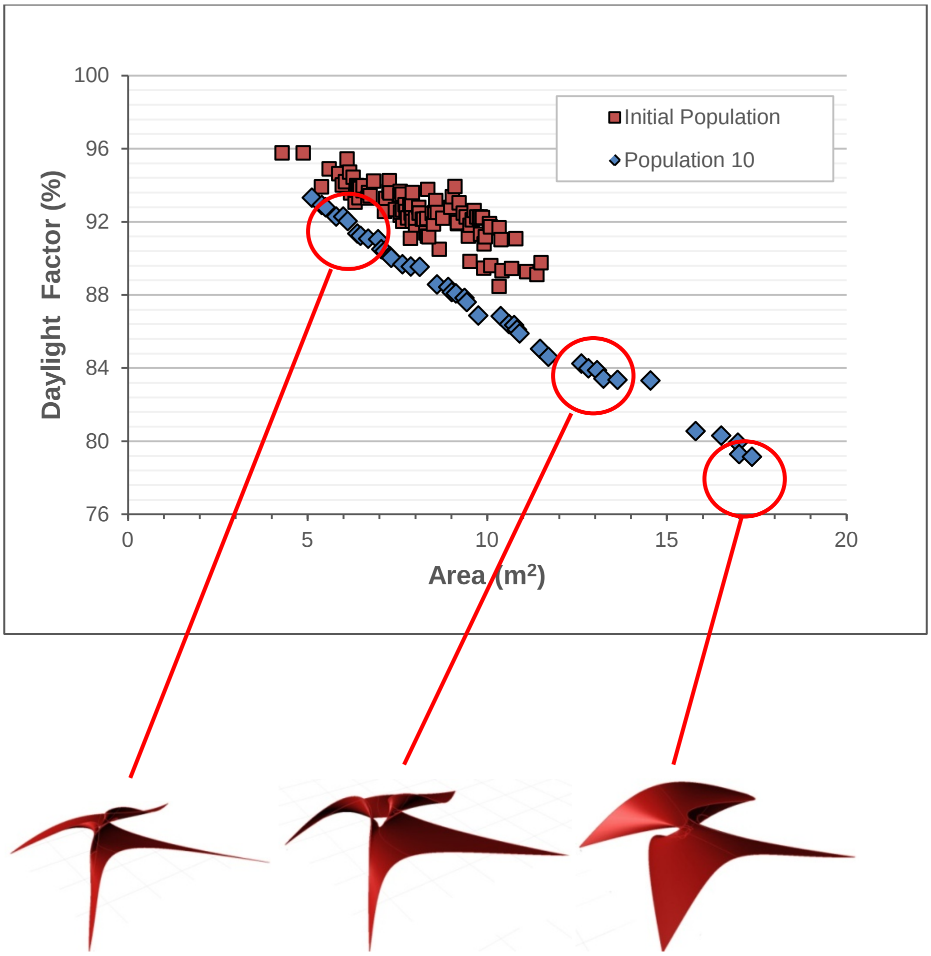


Fig. 28 – Second optimisation run with no restricted corner points: Pareto Optimum with some results.

Consequently another optimisation cycle was introduced in the optimisation process. This time, not only fixing the corner points as in the second case, but also the X and the Y coordinates (the coordinates in the border planes perpendicular to the ground plane) of all the points on the border of the roof structure were fixed: only the Z coordinates of those points were allowed to vary in the previously limited range (between 0.5 and 5 meters). In this case only 24 decision variables had to be taken into account. The results of this cycle of optimisation are shown in figure 29. These results, when compared to the results from the previous optimisation runs are visually much more consistent. A close look to the results from different parts of the Pareto frontier confirm visually that the top part of the Pareto frontier responds closer to one of the general objectives, lightness, a structure of this kind is clearly lighter because it uses less material, but on the other hand this structure will provide much less shade than solutions from the pool in the lower parts of the Pareto frontier.

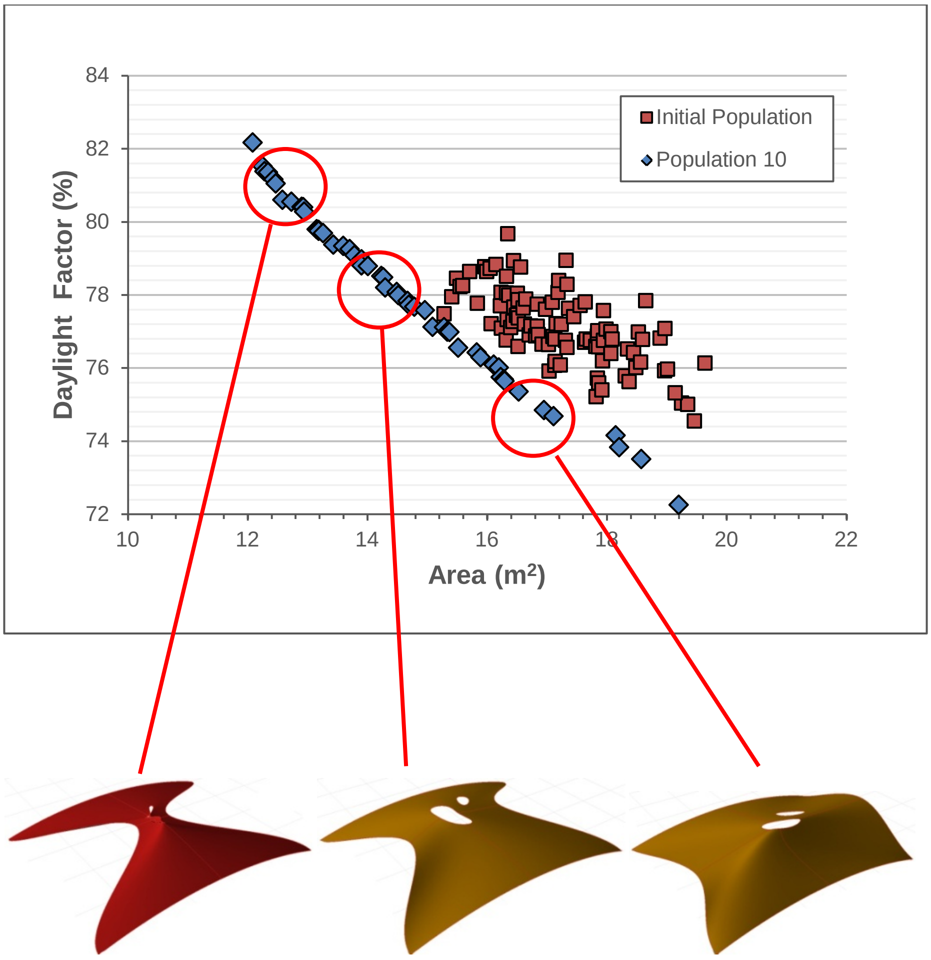


Fig. 29 – Third optimisation run with corner points and border points restricted: Pareto Optimum with some results.

After each of these optimisation runs, the designer was presented with a set of different geometrical solutions and their performance data. Eventually and if desired an extra or a different optimisation run could have been performed, if the results would not have been within reasonable and expected limits.

From this combined pool of results, all the non-dominant solutions of those three different optimisation runs were combined and grouped in a joint Pareto frontier (Fig. 30).

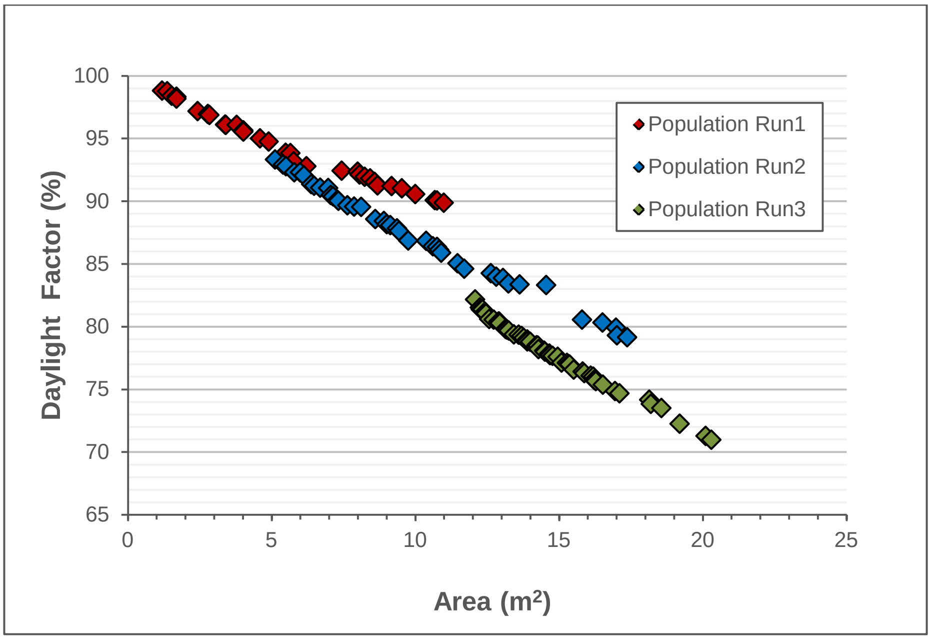


Fig. 30 – All instances together on the same graph.

This set of non-dominant solutions will then be used as the initial population for another optimisation run with the goal of producing a homogenous Pareto frontier which can cover all the different geometries proposed in the first runs (Fig. 30). One can observe that while the Pareto frontier in figure 31 is represented as a broken line with gaps, the Pareto frontier generated after this optimisation run with the combined results, is an improved and homogeneous curve.

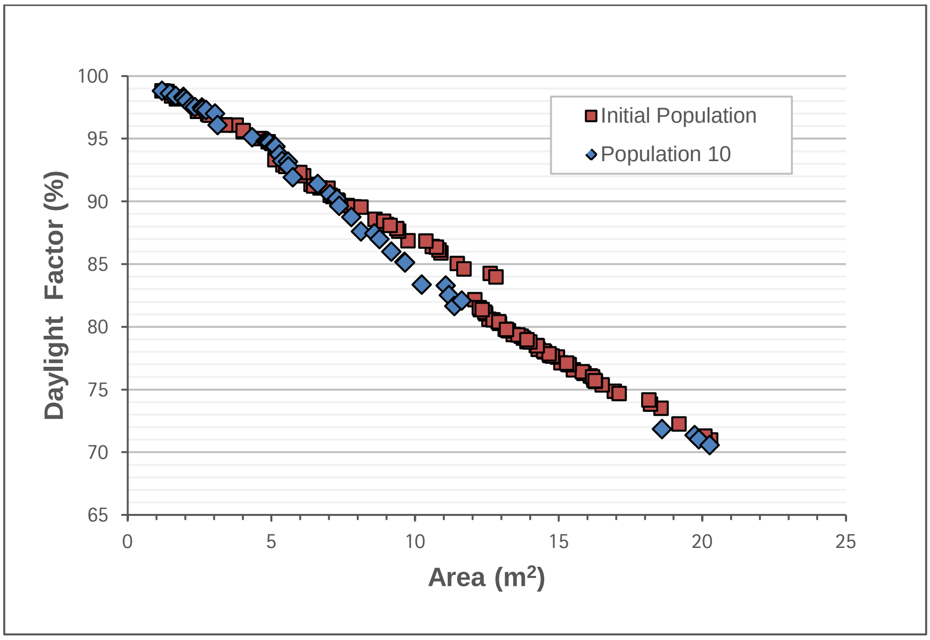


Fig. 31 – Global optimisation: initial population and non-dominant solutions after 10 generations.

From this pool of solutions, a set of desired outcomes was selected by the decision maker for no other reason than his personal choice (Fig. 32).

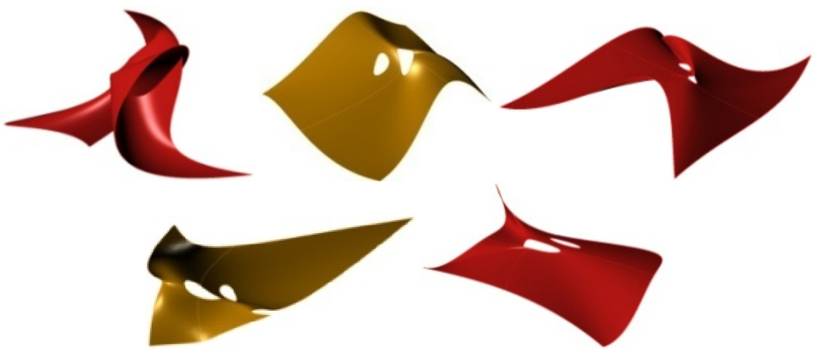


Fig. 32 – Designer's Choice for a final optimisation run with a weighting factor.

A closer analysis of the selection showed that all solutions were located more to the centre of the Pareto frontier. This apparent 'preference' was then translated into a set of weights of (0.5, 0.5) to be used in the decision making methodology. The non-dominant population of the last run was used as the initial population of this final optimisation run. The solutions are shown in figure 33, an as can be observed from the results of this optimisation cycle, better results are produced and the results all converge to the centre of the Pareto frontier, which is the region apparently preferred by the decision maker. In this way, the software construct will thus automatically generate possible solutions that are closer to the decision maker's preferences.

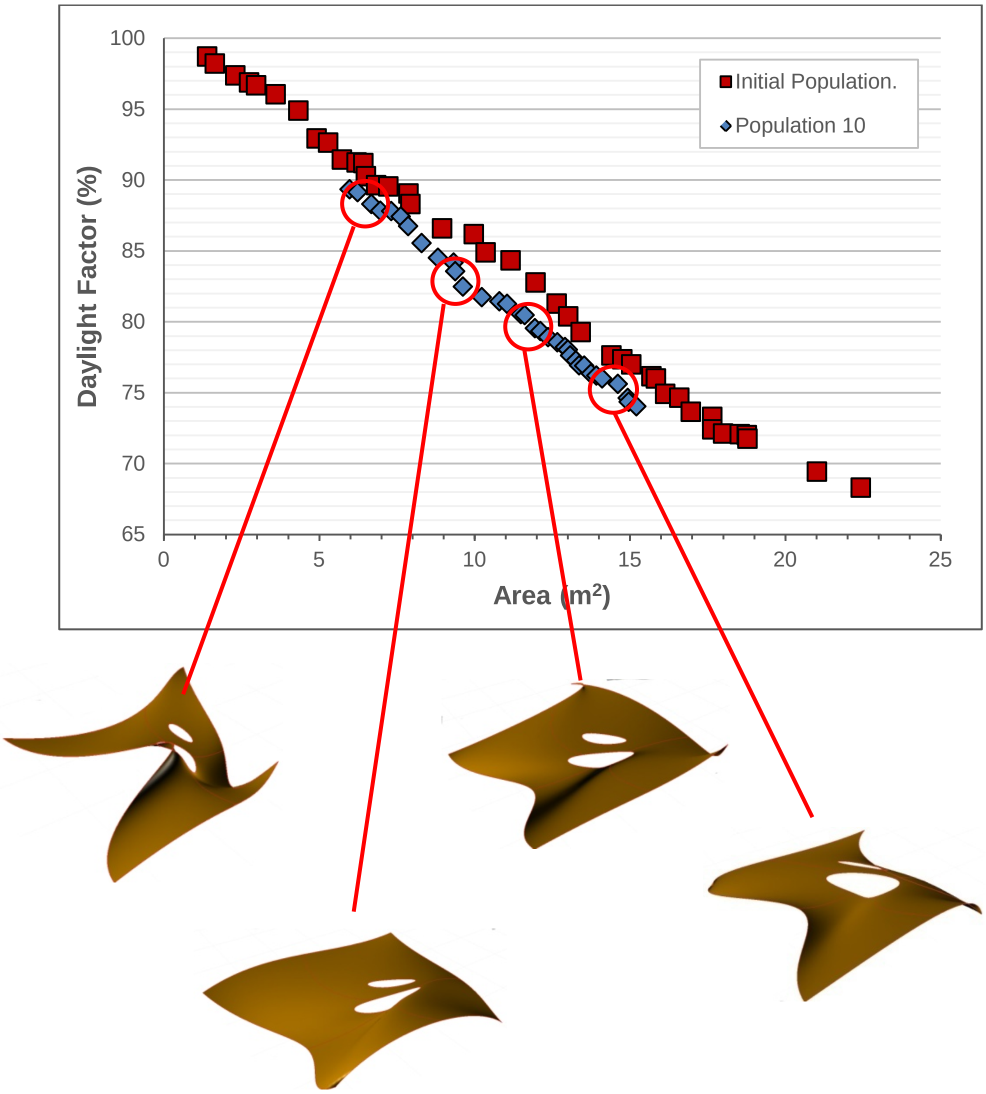


Fig. 33 – Final optimisation run with a weighting factor derived from the designer's choice.

4.1.6 Structural analysis with FEA

Although the structures which were calculated and optimised during this first case study are highly hypothetical, an attempt was made to comply with structural integrity by running the optimisation model with the inclusion of automated Finite Element Analysis (FEA)(Fig. 34).

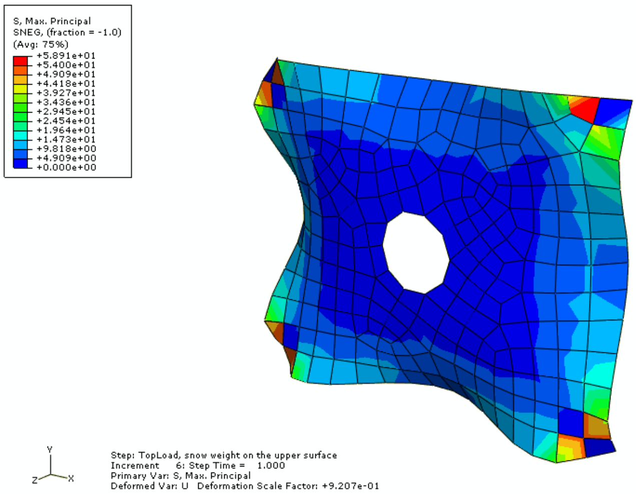


Fig. 34 – Surface with a fold (left) and surface with no fold (right).

Although overall test runs have shown that using the proposed construct, FEA can be integrated at this phase in a relatively straightforward manner, during testing two principal types of problems occurred. Firstly, the NURBS surface, which is controlled by changing the coordinates of the control points, can easily fold through itself which does not allow for a direct FEA, and should thus be avoided in this test case. As matter of fact, most of the time some part of the surface is folded, and only if one applies very big restrictions on the range of variation of the coordinates of the control points it is possible to avoid that the surface does not show any folding.

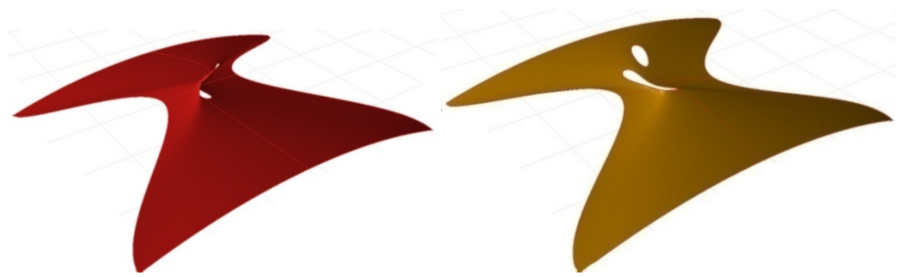


Fig. 35 – FEA analysis on a surface with no folds.

Extensive test runs were done, and eventually the surfaces were given a different colour to evaluate visually if folding occurred, yellow for non-folding surfaces and red for surfaces which did contain a fold (Fig. 35).

However, if the variation range of the coordinates was too restricted, only very predictable surfaces were generated, but on the other hand if the coordinates were less limited a considerable part of the generated surfaces could not be analysed in the FEA software and had to be rejected beforehand, without visual presentation to the designer. It was stated that in this first case study, form would prevail above structural integrity. The generated optimised roof structures are at the end only (rather precise) form suggestions. This should be usually enough exploration in the conceptual phase of the design process, and in any design process, if it was decided to continue with development. During and after the detailing process other simulation and analysis software can be applied, some with even better reliability than Ecotect, software such a ESP-r (2010) and EnergyPlus (2010), which can provide the architect with a more detailed analysis of the proposed object. However, as ESP-r and EnergyPlus are software programs which are not that simple to integrate in an automated optimisation cycle, they were not considered for this research work, but they are excellent tools for manual analysis and simulation and can be used to double check and validate the results from the optimisation construct.

4.1.7 Graphical User Interface

A typical GUI for an optimisation process, as was executed in case study 1, could be organised and function in the manner described and illustrated in this paragraph (Fig. 36). The basis parametric model which the designer will use for optimisation is presented, and from two (in this case) pop-up menus the designer can make a selection between different characteristics, which will influence the optimisation process. The material can be selected, in this demonstration the roof-like structure can be built in three kind of composite materials, and the total number of optimisation cycles which the optimisation algorithm will run before presenting results, can also be chosen, although the total number of optimisation cycles depends very much on the kind of optimisation problem (Fig. 37).

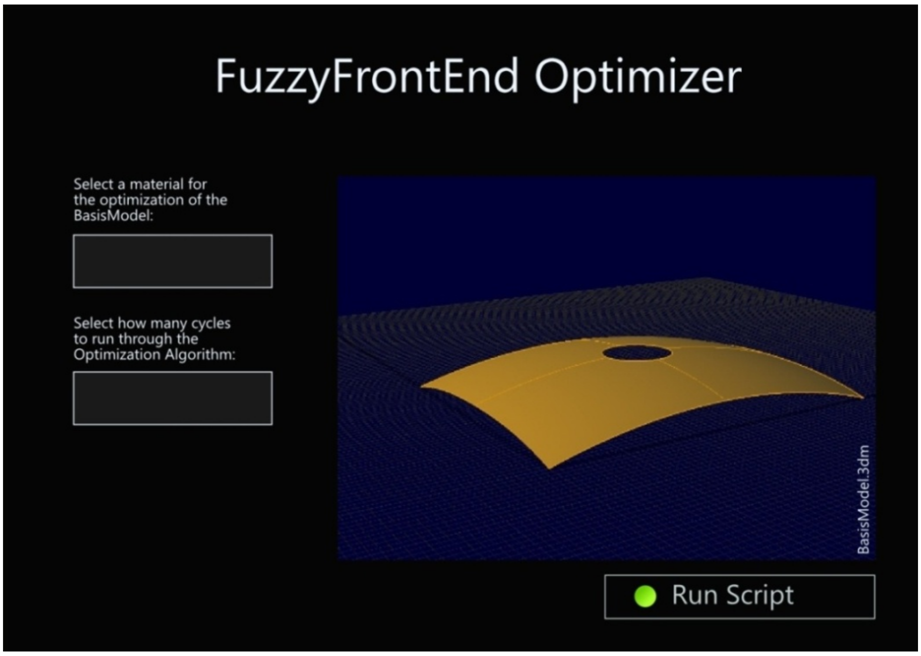


Fig. 36 – Opening Screen GUI.

Computational time is also an important part of an optimisation process and depends also in part also on the computational power of the computer where the software construct is running. However, it was argued that any parametric model, before any optimisation, should be tested for fitness. If the optimised results do not show an important difference with the original model, this could mean that very little can be expected from this digital design process. This is an important first iterative cycle in the software construct. In a fully digital design process, form generation and form finding are paramount to the positive evolution and final outcome of the process. But promising parametric geometries do not always generate interesting results or distinctive modification of form and appearance to qualify for optimisation as it is intended in a process of digital design. It is therefore important for the designer to understand when a digital design process has to be aborted and restarted with another better prepared basis model. Thus, with the intent to proof-run the process, in the proposed FFO the designer can choose from a pop-up menu how many generations will be created before presenting the optimised results.

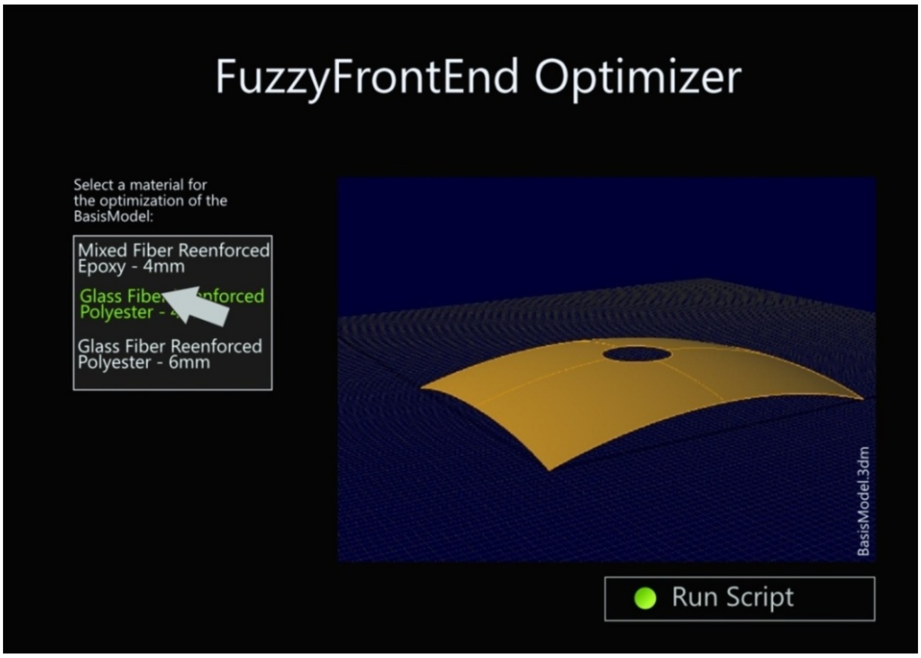


Fig. 37 – Selection of some of the parameters.

Once started running the selected quantity of generations, the optimisation algorithm will then produce an intermediate selection of six different Pareto-Optimum solutions evenly spread over the Pareto Frontier of best solutions.

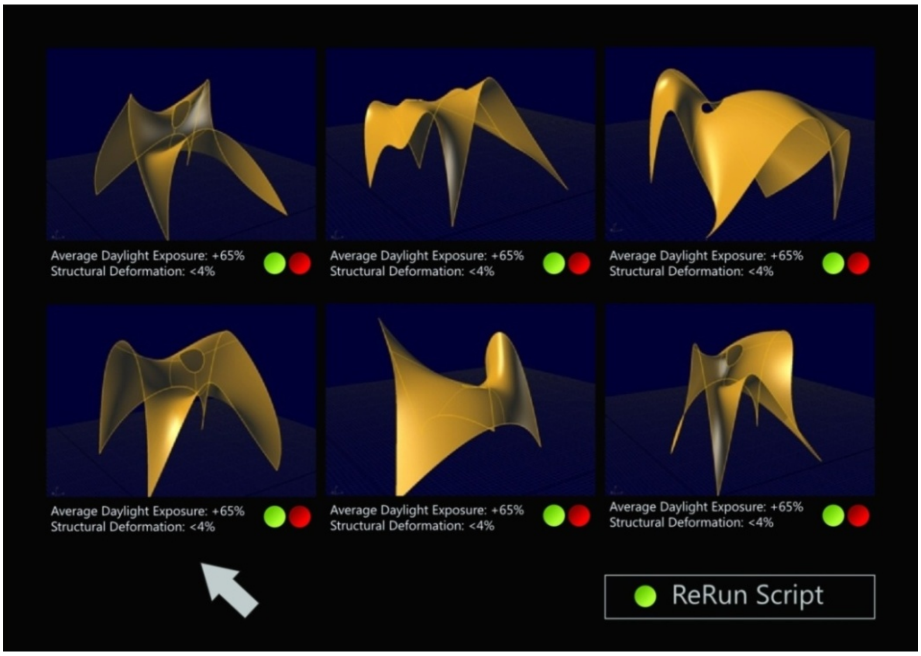


Fig. 38 – Presentation of selected Pareto Optimum solutions.

These solutions will be presented as images together with a small summary of some of the most important results of their specific performance simulation (Fig. 38). It will then depend on the designer which of the presented "promising" or "good" solutions will be selected and reseeded as phenotypes in the optimisation algorithm (Fig. 39) for another run.

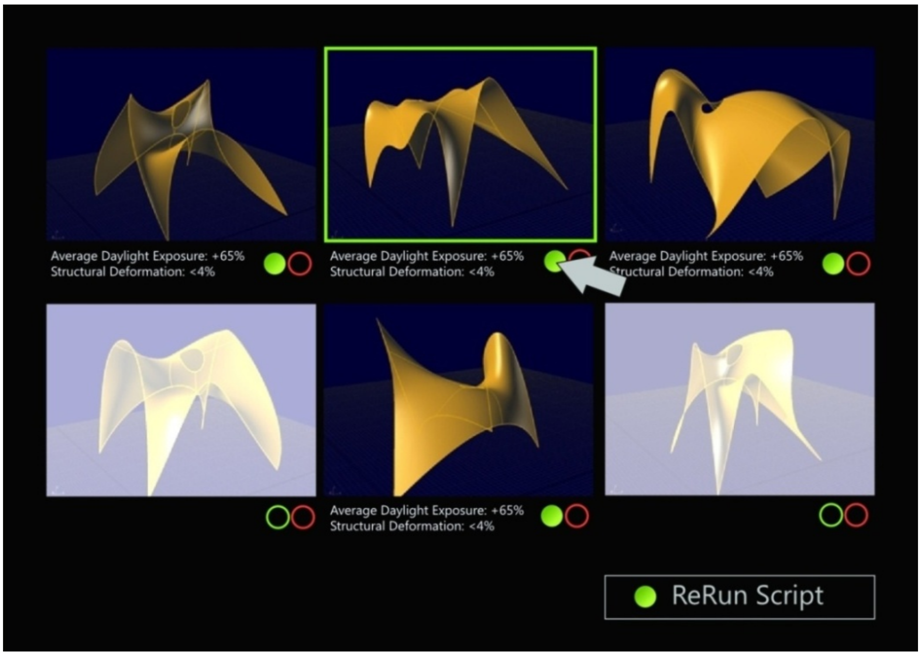


Fig. 39 – Selecting the preferred solutions for re-seeding.

Whenever one of the intermediate solutions seems like a promising result for development, more information can be provided and the designer can analyse in detail if this solution fits his expectations, after which he can either choose for further development in detail, or return to the optimisation iterations (Fig. 40).

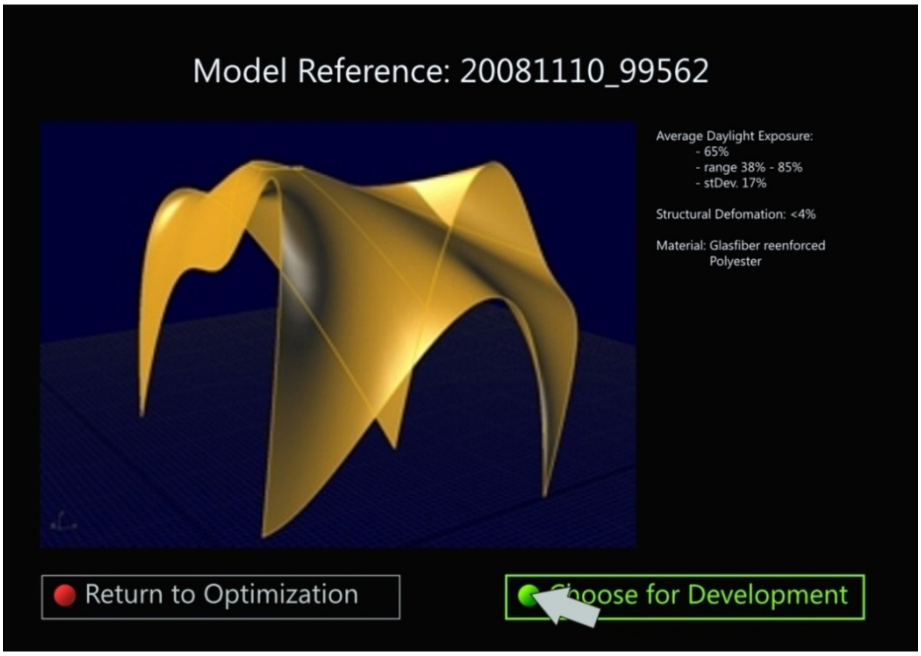


Fig. 40 – Analysing a possible candidate for further development.

This process can go on as long as no solutions are satisfying the aesthetical criteria of the designer or architect. Based on research on similar optimisation goals and with the use of the same MOEA, it is expected that, with a suitable initial geometry a fruitful process of form generation will produce valuable solutions after ten cycles at most (Gaspar-Cunha & Covas, 2004). Continuing the process of optimisation would only generate similar or imperceptibly different solutions unrelated to the goal of optimisation in design as intended in this study.

A graphical user interface is in no way indispensable in the process of optimised generative form finding or building. The process of optimisation can almost completely be executed in the background or in batch. However, it will make the process more cumbersome and with a different perception of interaction.

4.1.8 Conclusions

The resulting optimised design combines both quantitative and qualitative evaluation of the design's performance, leading to the exploration of a wider range of objectively better design solutions at an early stage in the concept phase. From the quantity of results produced during the different optimisation runs it is clear that careful manipulation and calculation of the necessary data for evaluation will be inherently cumbersome and difficult. The presented method in combination with a specific software construct, but adapted to a proposed general framework and software programs proved to be useful in the application of interactive optimisation of simple doom like structures.

A necessary and very important characteristic of the conceptual phase of the design process is the generation of information (see also 3.1.6), and this information provided to the designer through iterative processes and by presenting intermediate proposals and results which can be evaluated in short time. Decisions can be made at any part of the process to pursue or the restart the process completely or to return to a previous stage.

Different strategies for optimisation were applied and were combined. This is an important feature of the optimisation part of this software construct. The introduction of different weighting factors allows and contributes for a more targeted optimisation process and introduces iterative possibilities in this part of the process. Different boundary conditions were also applied with visually very clear results on the aesthetic outcome of the solutions. It became also clear that even some solutions at-first-sight impossible, and in a regular non digital design process probably immediately rejected, or not even considered for even minimal development, was introduced in the generative part of the optimisation process contributing this way to novel and valuable outcome.

So within the optimisation process, this tweaking of the stress factors together with adjusting (purposeful or playful) of some of the constraints and boundary conditions combined with an iterative attitude during the complete optimisation process can be recognised as tell-tale interactive behaviour.

As a final result of this first Case Study a Pareto frontier graph was presented depicting a set of optimised solutions, and, if the GUI is applied to the optimisation process, at the same time the decision maker has access to a 3D presentation of the objects and a summary of their performances results. This selective data provides the designer with enough information for him to pursue with the design process. Each one of the solutions out of this pool of best performing concepts can be used as the starting point for subsequent detailed design with less uncertainty about the future performance.

The positive final results of this Case Study did also validate and approve the fully functional capacities of the software construct. All the different parts, although written in different kinds of scripts with the use of different script languages, all the information about the object described in different digital formats and files proved, through carful importing and exporting between the different CAD and evaluation software, to maintain the essential and basic information until the end of the process and be ready for further exploration.

4.2 Case study 2

The first Case Study was prepared and executed to implement and test the proposed interactive optimisation method. Different CAD, simulation and analysing software were used to validate a software construct with the ability to trustfully exchange information between the different components and produce the desired results in the form of an optimised geometry according to pre-established objectives and design intent.

In this second Case Study the process of interactive optimisation will be demonstrated using a practical real world design problem. The objective of this experiment is to study the influence of different shading devices on the natural lighting and on the thermal behaviour inside an enclosed cube. The proposed interaction during optimisation will be tested and demonstrated and general knowledge about streamlining the whole process will be obtained.General Experimental Set Up

A test cell where the potential of low span membranes can be explored was constructed at the University of Minho, on its Campus de Azurém, Guimarães, Portugal. Its concept was presented in detail in previous publications

(Mendonça, 2010) (Reis, 2011). This prototype is composed of a cube with 2400 x 2400 x 2400 mm (Fig.41). Its main structure is made of aluminium profiles. The west and east façades are made of an opaque white polyester/PVC membrane inserted into the aluminium profiles by a PVC rod. Its structural stability is assured by four 20cm long steel rods which are compressed against the membrane by two crossed steel cables fixed to the corners, that also assure the cross stabilisation of the panels. South and North façades have transparent PVC as an impermeable layer. The north façade has a double layer pneumatic cushion system and the south façade has a single layer plastic membrane. That was left free for lightweight façade shading systems testing. In two previous set-ups, the cube was tested without a shading device and tested with a shading device consisting of a single membrane of open weave polyester PVC (Mendonça, 2010).

Fig. 41 – Membrane test Cell at the University of Minho, Portugal (Mendonça, 2010).

The objective of the proposed experiment is to study the influence of different geometries of shading devices built in lightweight materials on the thermal behaviour and the natural lighting inside the enclosed cube, based on the same shading coefficient of the shading device that can be seen in figure 33. The advantages of the shading device shown in figure 33 were reflected in the thermal performance, with a reduction verified on the maximum temperature and thermal lag (Reis, 2011). Apart from the thermal performance improvement, another important aspect is the better compatibility between thermal comfort and natural lighting that the shading device assures. The existence of this Test Cell presents the advantage that it will be possible to physically measure at least those two variables on different complex geometries.

To generate complex geometries typically 3D Cad software is used. Traditional 3D CAD software is based on simple geometric entities with just enough information encoded for the software to be able to create a 3D model within their own graphical user interface and prepared for application of the tools and operations used by this particular software. Therefore a geometry built in CAD software such as Rhino 3D does not necessary contain the exact material and spatial information required by building simulation and analysis software for detailed thermal studies. Performance analysis software such as EcoTect 2011 needs a different kind of information than that provided by a modelling software such as Rhino 3D to be able to calculate and analyse thermal, lighting and acoustic performance. In order to be able to present reliable simulation results it is important that the virtual 3D model complies with some basic rules for correct setup to perform thermal analysis. EcoTect 2011, the thermal simulation and analysis software used in this study, needs a 'closed space' with a specific set of internal conditions to be able to create thermal models. If the virtual setup complies with those basic setting it will be possible to perform a myriad of different analyses.

External shading devices have a direct impact on the perception of comfort inside a building. For one, they can regulate the amount of solar radiation which is allowed to pass through the window. Different shading devices are expected to result in different values for direct solar gains, one of the components of thermal sustainability, resulting in thermal variations inside the room. The shading devices used in this study were generated in Rhino 3D CAD software, transformed to 3ds file format and prepared for import directly in the Ecotect software. The shading device is set as a non-thermal zone with the proper material assigned to it.

In this experiment the only part of the setup which will be different for each performance analysis is the external shading device, and although EcoTect does allow for the importation of different geometries into an existing setup, it is neither necessary nor wise to build a full geometry, cube and shading device, in Rhino and then import this geometry in EcoTect 2011 each time one would like to perform a thermal analysis with a different shading device. The better option would be to build the basic setup of the test cube in the CAD interface of EcoTect 2011 and build a thermal model complete with all the necessary information such as orientation (window to the south), weather file (weather data for Guimarães) and the material assigned to the walls, the roof, the floor and the window. With this procedure one can start EcoTect 2011 always with an identical, fully prepared testing cube and simply import a 3ds-file of the shading device into the existing model, thus saving important processing time (Fig. 42).

Important for any simulation software is the use of validated weather data. The cube is placed in Guimarães, Portugal on the location of the Campus of Azurém at the Minho University. International databases such as the database from the US Department of Energy (Energyplus2010) provide reliable information about weather for a lot of countries and towns, but not for Guimarães. And weather data for this location is also not readily available from the standard database included in the Ecotect 2011 software package and thus had to be adapted from other sources. The weather data used in this study was compiled with the use of the Ecotect Weather tool and transformed to the necessary file format used by Ecotect 2011. Weather Data for Guimarães, Portugal were registered and compiled locally at the University of Minho during the year 2003.

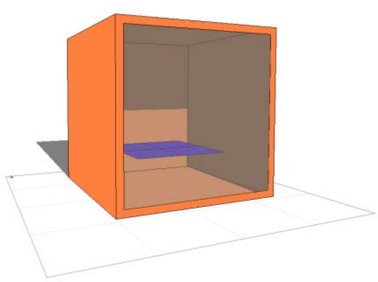


Fig. 42 – Basic Configuration of the CS2 Test Cube in Autodesk Ecotect 2011.

Parameters and Specifications – As mentioned before the model which will be used for simulation and analyses is a cube with the dimensions of 2400mm x

2400mm x 2400mm. The base cube was built in the CAD interface of Ecotect 2011 with the goal of including all the correct material properties, the necessary operational and spatial information, required to perform a suitable and detailed thermal and lighting simulation and analysis. The cube itself was set as a closed 'Thermal Zone'. To all the objects of this thermal zone - the different faces of the cube - were assigned materials with the same material properties, as will be described in the next paragraphs. Only the side of the cube facing south has an opening, completely filled by a centred window with the dimension of 2200x2200mm, in analogy to the test cell constructed on the campus of the University of Minho.

EcoTect 2011 can perform a thermal analysis as well as lighting analysis over an "Analysis Grid". For this experimental setup, a two dimensional analysis grid plane was considered with a dimension of 1400x1400mm on a height of 600mm in the centre of the cube and parallel to the floor plane. The quantity and quality of natural lighting in an enclosed space is calculated on this plane, as informed by the rules of *daylighting design strategies* for Ecotect, slightly lower than the standard height of a working surface such as a desk or a table (Ecotect Community Wiki 2007).

The analysis grid was divided in 3x3 cells which allowed for 9 nodes where specific lighting data could be calculated. The analysis grid can be divided and configured with as many grid cells as necessary and desirable. In this case studying the nine nodes do not compromise calculation time and can give feedback about the influences of asymmetrical shading devices on the values of the Daylight Factor. And although the variable for optimisation is the average value between the values of the nine nodes, it was considered helpful to be able to demonstrate that an irregular form of the shading devices influenced the even distribution of light inside the cube.

Materials - For the first simulation and analysis tests new "virtual" materials were created and added to the materials library of the software. These virtual materials were created, based on combined characteristics and parameters of other existing virtual materials in the materials library. Those new materials were created specifically for the calculation of the data of this experiment. To the walls, the roof and the floor of this cube a virtual material, "TEST 09122010 Wall", was attributed with the thermal characteristics equal to the material "Rammed Earth 500mm" from the standard material library of Ecotect 2011. The very large thermal mass of the material "Rammed Earth", provides enough inertia against fluctuations of the temperature inside the cube as a direct result of temperature fluctuations on the outside. Any change or lack of change in inside temperature could thus be attributed to direct solar gains through the window, with or without the external shading device. Further, and although "Rammed Earth" needs a considerable thickness in order to comply with those thermal mass characteristics, in this case about 500mm, it is possible in a virtual setting to reduce the thickness to only 10mm and produce the same reliable results. This way one can avoid interference with the calculation of "Adjancy" and Inter Zonal Gains, which calculates the influence of other thermal zones on the thermal behaviour of the space in observation and which is a feature of this thermal analysis and simulation software which contributes to its accuracy and its flexibility, but which in this case and in this study will not have any influence on the results, because calculations are only subject to one single closed space. Thus, in order to reduce computation time, material thickness was set to 10mm, in the physical world impossible to achieve, but in virtual simulations perfectly feasible.

For the window on the south side of the cube, the same strategy was applied. Again a new virtual material was created, named "TEST 09122010 Window", based on the characteristics of "Double Glazed_LowE_AlumFrame" material from the standard Ecotect 2011 material library (although any other standard material could be used). For this new virtual material only four characteristics were altered and set to almost maximum value of 0.99, Solar Heat Gain Coefficient, Alt Solar Gain (Heavywt), Alt Solar Gain (Lightwt) and Visual Transmittance. All other values directly related to the thermal behaviour of this material (U-Value, Admittance and Refractive Index) are kept equal to the values of the EcoTect 2011 standard material "Double Glazed_LowE_AlumFrame". In the simulation and analyses software used, Direct Solar Gains can be presented in a variety of graphs and tables. One very useful table is the relative contribution in percentage of DSG to the overall heating of the enclosed space. However, if the other factors which contribute to thermal comfort are relatively big, small differences between shading devices can only be observed in hundreds or thousands of a percentage point. Therefore the parameters of those materials which showed less variation in relation to the outside air temperature were introduced in the simulation software. With this virtual configuration, the results of the DSG calculation allowed for a bigger amplitude, thus augmenting its reliability to distinguish between different solutions simulated.

Those first tests were successfully executed with the thermal zone conditions completely featured out (0 value for all the settings). But these settings do not simulate real world conditions and for the optimisation experiment all the parameters and requirements were set to common sense values and normal behaviour of a living person in a room. The results of these Thermal Zone Settings however will only be visible in the Passive Gains Breakdown analysis, but will not influence the item of Passive Gains which is of interest in this study because it is this particular characteristic of the thermal behaviour of a building – Direct Solar Gains - which is directly influenced by the presence and the geometry of a shading device, and thus the setup allows for a direct "measure" of the effectiveness of solely the shading device, which is the purpose of this study.

The shading devices need to be lightweight and at the same time reliable and strong structures. Composite materials and in particular Fibre Glass Reinforced Polyester, is one of those materials with the right properties to build shading devices with the geometry as the ones proposed in this study. The reflective properties of materials used in the construction of the shading device can and do influence the thermal behaviour of the shaded space, and the thermal simulation and analysis software does take characteristics of the material and the colour in account for the calculations. Since the shading device developed in this study was not built at the moment of execution of this simulation - the material used as a standard for the calculations is a simple white generic plastic, with characteristics very similar to a glass fibre composite shading device with a white gel coat finish. This latter material has the most, probable characteristics of a shading device constructed for testing in real world settings.

Simulation Parameters - In order to be able to execute a thermal analysis, Ecotect 2011 needs to perform an Inter-Zonal Adjacencies calculation, which takes in account the potential effect of the thermal behaviour of one zone on its neighbouring zones' thermal behaviour. Since this experimental setup only has one Thermal Zone, eventual Inter-Zonal Adjacencies do not have any influence on the kind of calculations which are pretended for the analysis, but always have to be done before any thermal analysis. As has been stated before the calculation of this Inter-Zonal Adjacencies is time consuming. The MOEA optimisation algorithm needs to calculate these Inter-Zonal Adjacencies for each new shading device which has to be analysed and the complexity of those shading devices is directly related to the computation time necessary for the calculation. Therefore different parameters are all set for maximum speed of execution. The first parameter, Sample Grid Size, relates to the accuracy of the calculations and is set to the maximum value of 2500, the least possible accuracy. The Adjacency Tolerance is set to the very low value of only 1 mm. This low tolerance together with the distance of 50mm at which the shading devices are mounted in front of the south facing window of the test cube, avoids inclusion of the complex mesh geometry of the shading devices in the calculations, considerably speeding up computation time.

4.2.1 Shading Devices

Shading devices are but one element in a strategy of passive solar design which uses the sun's energy to regulate the thermal comfort in spaces and the sun's light to illuminate. Passive solar design applies simple and cost-effective systems with little or minimal maintenance, and takes advantage of the characteristics of materials and specific configurations to balance energy consumption in buildings (Fig. 43).

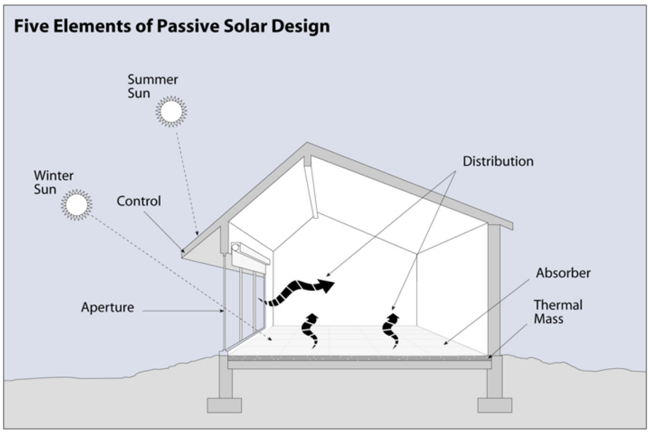


Fig. 43 – Passive Solar Design (Energy Savers, 2011)

This Case Study is about optimising shading devices and as mentioned in chapter three, one of the first tasks of the designer in this process of interactive optimisation is to propose and define a proper parametric geometry which allows for creative exploration and which can be optimised according to acceptable constraints and design intent. For this reason different VBScripts were developed to generate and explore possible solutions for decorative shading devices which would allow enough parametric freedom and which could be coupled to the proposed variables for optimisation.

The traditional tool used by designers and architects for (not only visual) exploration is sketching (Johnson, 2011). However, in a digital design process, initial exploration has to be done by different processes, for example, simple scripts written in the CAD's scripting language can be used for a similar process of probing the solutions space to get a better understanding of the general design problem. In this study Grasshopper, a modular graphic scripting tool, and RhinoScript, a more classical approach was applied for quick exploration of generative geometries (Fig. 44-48).

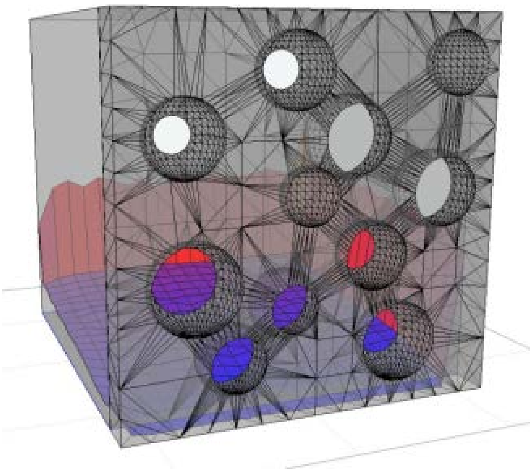


Fig. 44 – Parametric exploration of shading devices using random placed spherical cut-outs with a maximum surface opening between 40% and 60%.

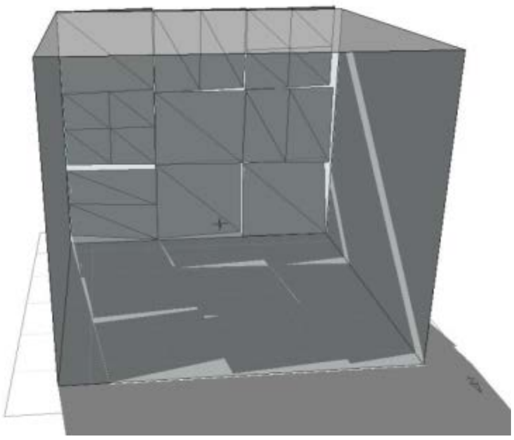


Fig. 45 – Parametric exploration of shading devices with tilted panelling.

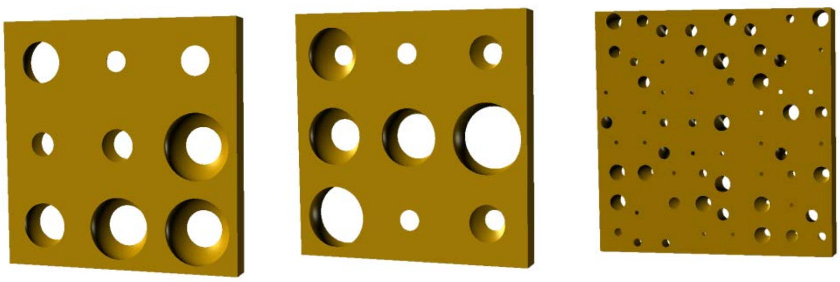


Fig. 46 – Parametric exploration of shading devices with conical windows.

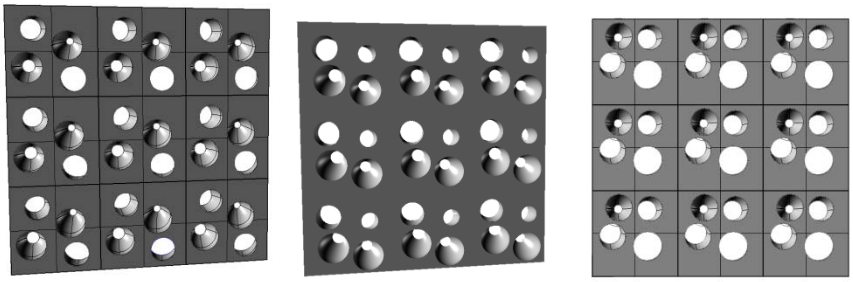


Fig. 47 – Parametric exploration of shading devices with patterned lofted windows.

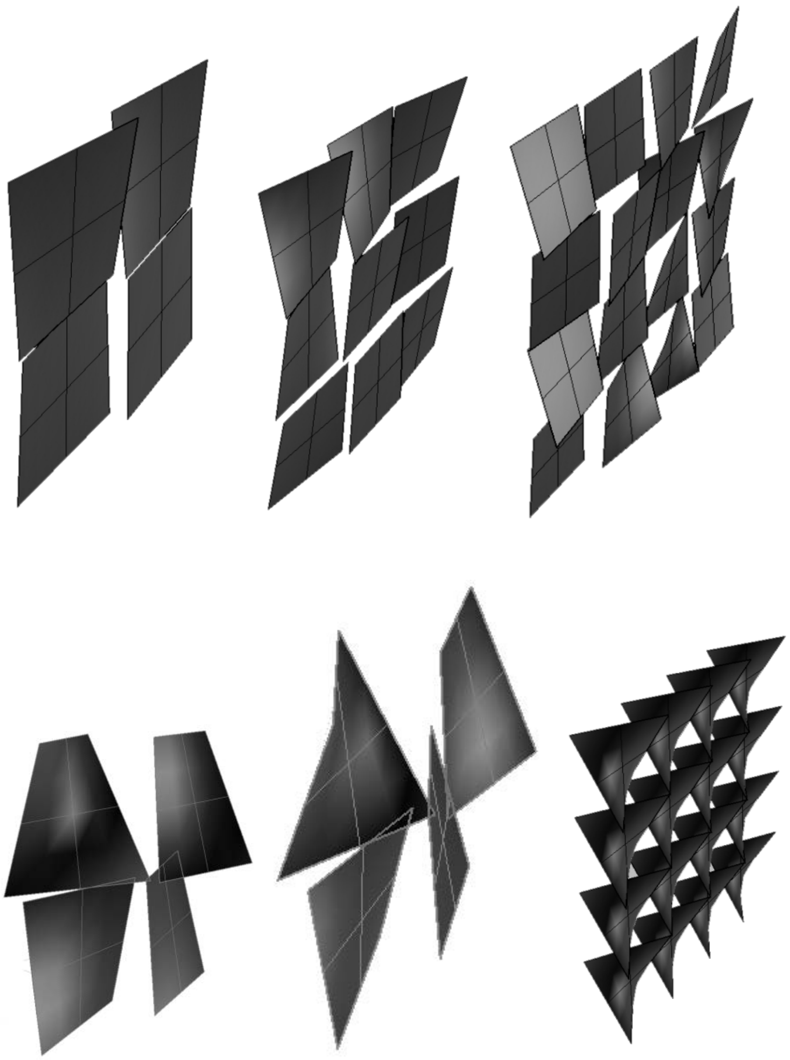


Fig. 48 – Parametric exploration of shading devices using basic tensile structures.

After this quick exploration, the final scripts for this case study written in RhinoScript and were used for exploring initial ideas and to verify if the generative algorithms comply with a digital design process, but also to verify that they were flexible enough for optimisation. As can be expected - and can be considered as a perfectly normal part and procedure of a digital design process (see also paragraph 3.2.3) - not many of the results of this initial exploration of possible solutions could be used for optimisation. Some of the proposals were not suitable as a shading device at all. The use of simple tensile structures (Fig. 48) was thoroughly explored. Different scripts were written to build tensile structures with different shapes, with different sizes, combination of smaller and bigger elements, etc. But, contrarily to expectations, the design of this kind of shading device could not be transformed in a practical and manageable process for optimisation within the limits of this study and the available computational power accessible at this moment. In order to create valuable shading device using tensile structures, more complexity in geometry, configuration and construction is needed and consequently the computation time involved in such a process requires more powerful processes – an issue which is readily overcome in the future.

The use of conical “windows” with a relative joint aperture of between 40% and 60% of the total surface area of the facade, such as shown in figure 44, with conical windows or with a random distribution of the conical windows (Fig. 46), with a pattern like distribution of lofted windows (Fig. 47), did also prove to generate solutions with too little visual and aesthetical differences and thus with no reasonable necessity for optimisation.

A plausible option proved to be a grid of cone like objects, and in analogy to Case Study 1, a script in RhinoScript Visual Basic was developed in the CAD software Rhinoceros 3D to generate different structures of cone-like shading devices and explore the parametric possibilities of this geometry (Fig. 49). Dozens of different shading devices were generated with different forms, depths and openings at the front, ultimately limited to a structure of nine identical cone-like forms or geometries.

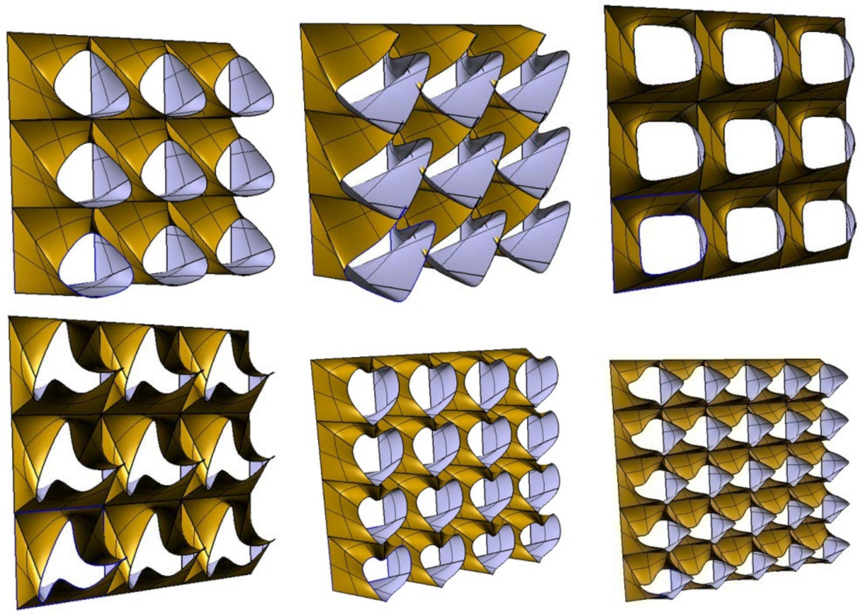


Fig. 49 – A collection of different shading devices generated with the same script.

As mentioned before, Autodesk Ecotect 2011 is a powerful and complex software tool for the simulation of the thermal and solar behaviour of buildings and allows for the representation of the results in a myriad of different graphs, tables and combinations of both. Many simulated measurements can be used as an indicator of "comfort" in a closed space and, consequently, of the successful application of a shading device.

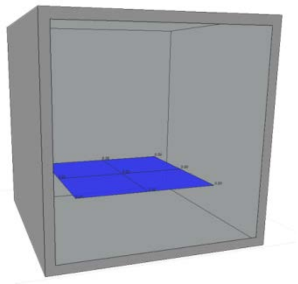
However, due to the particularities of the software and the graphical layout of the user interface, interpretation of the results are mostly based on visual interaction with the designer in real time and are therefore difficult to be executed autonomously by a script. On the other hand some of the results generated by this manually applied simulation did not provide significantly different results and without a logical understanding for the reason of this similarity it seems difficult to justify which parameters can be used for optimisation, since changes in thermal comfort have to be directly related to changes in the geometry of the shading device.

Therefore, it was considered necessary to test some possibly useful variables in the process of lighting and thermal analyses with the goal of finding significant variables for optimisation. Figure 50 shows the comparative results of a manually calculated simulation of four different configurations, a window without any shading devices (configuration #1), a window with a wall in front (configuration #2) and two configurations with different shading devices. The goal of these calculations was to test if the objectives for optimisation did correspond to the expected values and if generated results demonstrated useful differences. Four different variables were calculated (see also annex B) and the values for a point in the centre of the cube at a height of 600mm, represented and compared in a graph (Fig.50).

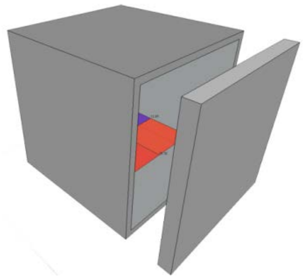
The Daylight factor is the ratio between the available illumination indoors at a certain point in relation to the available illumination outdoors on a cloudy winter day and the Direct Solar Gains, which is directly related to the temperature increase in that particular space, presented the most interesting results: resulting in most significant differences between various solutions. Furthermore, both parameters are directly related to the existence and the configuration of shading devices, and both parameters are variables which need to be optimised in a building strategy for passive solar design. For the purpose of objective optimisation, natural daylight needs to be maximised at all time while direct solar gains in winter has to be maximised and in summer minimised. From the tables in figure 50 it is clear that Daylight Factor and Direct Solar Gains are higher for shading device #3 with shorter cones and thus a more 'open' configuration, than for shading device #4 with deeper cones and thus less solar rays entering the space. The results are consistent with what could be expected and the results showed that the proposed objectives, Daylight Factor as well as Direct Solar Gains, can be used as variables for optimisation.

But while a Daylight Factor can be calculated at a specific point somewhere in space, and the exact locations for measurements of those points are determined by local regulations, Overall Iluminance which describes the amount of light in Lux hitting a specific point or a node in a measuring grid, are more appropriate for the understanding of the homogeneity of the distribution of light and the penetration of light in a deep room or space. While this particular objective could also be used as a variable for optimisation, the dimensions of the cube used in this experimental set up is not at all deep enough for significantly differing results – see also the discussion on this subject later on in this paragraph - thus the geometry of the shading device could provoke unwanted results and side track solutions which otherwise could be worthwhile considering. Mean Radiant Temperature as a variable for optimisation, although with significant differences between the four configurations, has to be rejected for the same reason. While Mean Radiant Temperature is very important for accessing comfort in a space, calculations depend significantly on many more variables, such as colour of the walls and colour of the shading device (which are not taken in consideration in this study), and it is therefore unclear how and at what order the geometry of different shading devices can contribute to the final result.

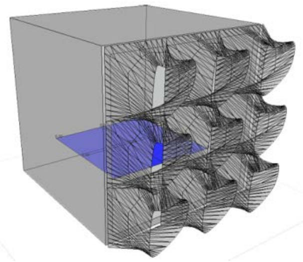
On first sight, it is not intuitively clear in all cases why different kinds of analyses of the virtual simulations generate unexpected results such as can be observed comparing Mean Radiant Temperature (Fig. 50). According to Jon Gardzelewski (personal communication, January 10, 2010), head of Freeform Energy and an expert trained in the application of lighting and thermal analyses of buildings, in this case the dimensions of the cube are too small for some of the analyses to show significant or reliable results. If the dimensions of the space are too small the results in the middle of the cube are too much influenced by the edges and some of the variables calculated in the simulation do not demonstrate expected results.



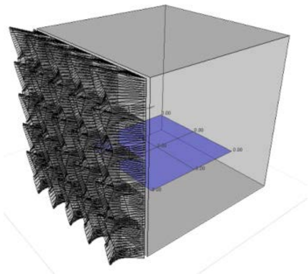
NO SHADING DEVICE



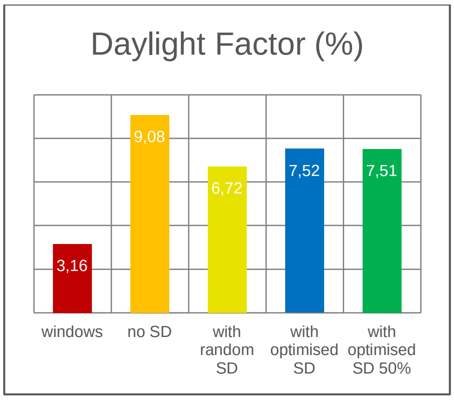
WITH A WALL IN FRONT

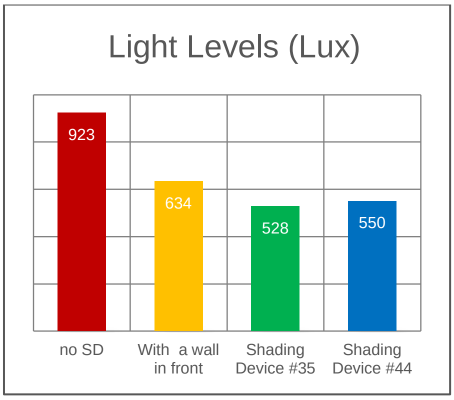


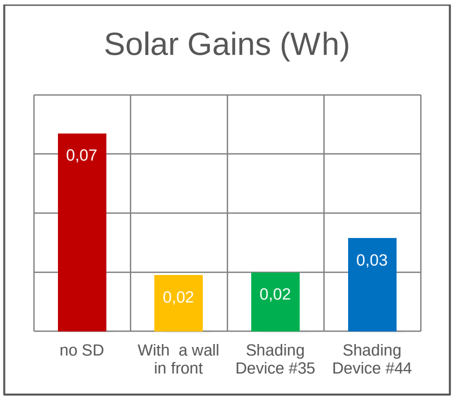
SHADING DEVICE Nº 35



SHADING DEVICE Nº 44







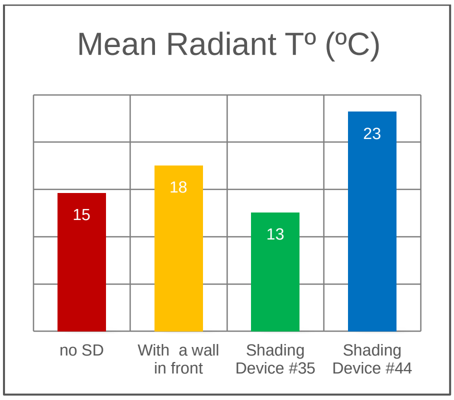


Fig. 50 – Selection of some of the simulation results tested.

Thus, it was concluded that most discriminating results were obtained by comparing the results of the calculation of the Daylight Factor (which includes the calculation of the Sky Component in its formula) of the different configurations, and also other reliable results can be obtained by comparing the values of the calculation of the Direct Solar Gains on only two specific but significant days of the year. Both characteristics can be used to understand the efficiency of the shading devices.

4.2.2 Analysing Shading Devices

Optimisation is still a time consuming operation and therefore, in order to save valuable time, before starting with the process it is necessary to verify manually if generation, simulation and analysis of the proposed kind of object is possible, plausible and generates significant results. For this reason two shading devices were selected for analysis based on expected differences in results (Fig. 51).

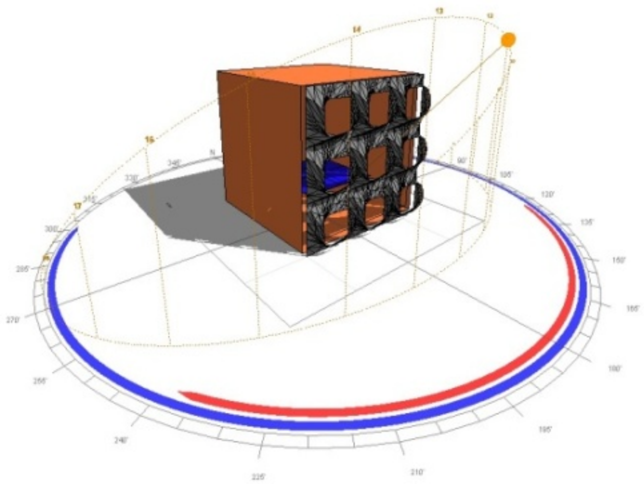


Fig. 51 – Two different shading cones with expected different results, #1 (top) and #2 (bottom).

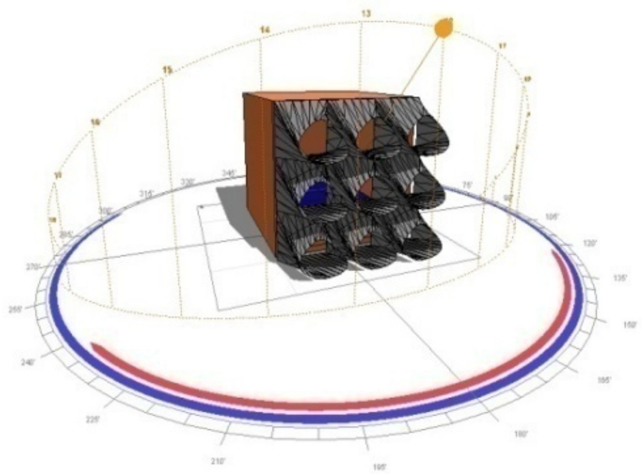


Fig. 51(cont.) – Two different shading cones with expected different results, #1 (top) and #2 (bottom).

The first proposal selected for testing was a construction of nine shorter shading cones, where the parametric solution resulted in a wider opening at the front part of the cone, about 400mm and with a total depth of only about a quarter of the width of one module (800mm), in this case about 200mm (Fig. 51 top). In this setup direct sunlight can enter the cube in winter time as well as in summer time. Considerable direct solar gains are expected to be noticeable in the calculated data. In the second setup (Fig. 51 bottom), with a smaller opening at the top and with a depth of 800mm, equal to the width of one element, direct sunlight is expected to be totally blocked in summer but allowed to enter the enclosed space of the cube in wintertime. The results of direct solar gains calculated over the year are thus expected to be different than in the previous setup, with an expected bias to gains in wintertime. If one can notice a difference in direct solar gains between the different shading devices, direct solar gains can be one of the objectives to optimise in this case study. If one can reliably measure those two variables one can start with the integration of the different software with the goal of preparing for interactive optimisation.

In order to validate the results with a real world setup, a third set of data will be generated with the same shading device as in the first setup, but with four of the nine “cones” completely blocked at the back (Fig. 52 top). For benchmarking purposes, all validations will also be executed on the same standard cube, but without any shading device (SD). Finally, for verification purposes, another setup will also be tested, checking only for direct solar gains. In this setup the shading device is the same as in first setup, but will be connected directly to the cube (Fig. 52 bottom), different from all other setups where the shading device is mounted at a small distance in front of the cube. The results will also be presented, although it is expected that this aperture only has a minor or even no influence at all.

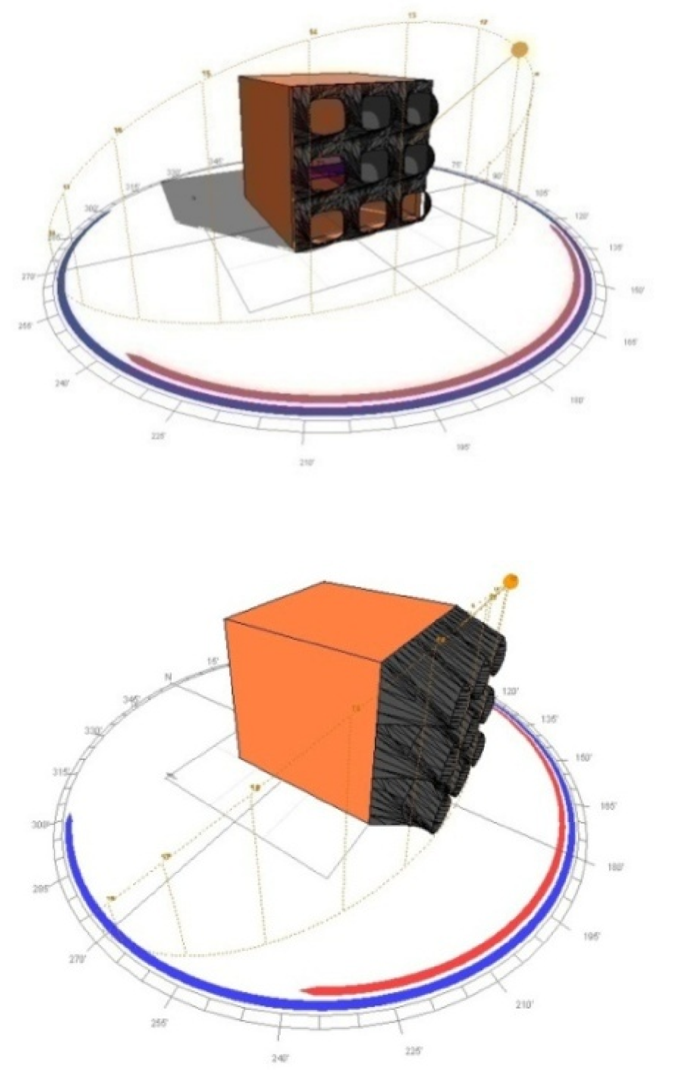


Fig. 52 – Test Cube with Blocked SD (top) and Test Cube with fixed SD (bottom).

Two variables will be studied with direct influence on the results of optimum passive solar design. A shading device that will be placed directly in front of a window does create a conflicting situation. On one side one would expect the biggest possible window size (maximum aperture of the shading devices) to allow for maximum natural lighting and higher comfort and improved living experience inside - as a matter of fact a shading device is always an obstruction. On the other side one needs to optimise the use of direct solar gains by reducing direct solar radiation in the summer months, and thus reducing the need for cooling the space, and maximising solar radiation in the winter months, equally reducing the need for heating.

EcoTect 2011 calculates the daylight factor according to the Building Establishment Split-Flux method (Daylight Factors: ECOTECT), a formula for the calculation of the Daylight Factor introduced by the Building Research Establishment and used by Ecotect. This method assumes that the quantity of natural light which reaches any point inside a building, and ignoring direct sunlight, is the sum of three components: Externally Reflected light from objects on the outside of the building such as other buildings or trees, Internally Reflected light from surfaces within the building, and the Sky Component which refers to the light that reaches any point inside a building directly from the sky through an opening such as a window. This experimental setup does not contemplate any outside objects, and does not contain any objects inside the test cube nor is intended to take any characteristics from the building material into consideration for this analysis. It has also to be reminded that in order to obtain significant and valuable results and feedback from the BRE method of daylight factor calculation, the cube is too small in size. The distribution of light inside such a small space is simply too homogenous. So instead of calculating Daylight Factor or Daylighting one can concentrate on the Sky Component only (Fig. 53).

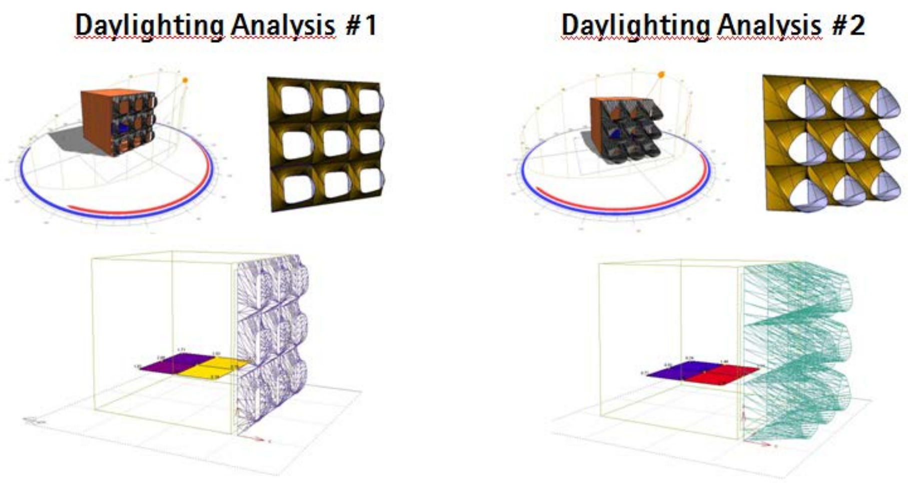


Fig. 53 – Sky Component over grid Shading Device #1 (left) and #2 (right).

Thus, it was decided to calculate the Sky Component over a grid of nine nodes executed with the following parameters: Natural Light Levels, Over the Analysis Grid, Sky Iluminance calculated automatically from the model latitude, CIE Overcast Sky condition, clean window and with the method of calculation set to the Regular Compliance Method. Results of the selected nodes are shown in figure 54. Significant differences can be observed between the results of both shading devices demonstrating a clear influence of the depth of the shading device on the available light inside the cube.

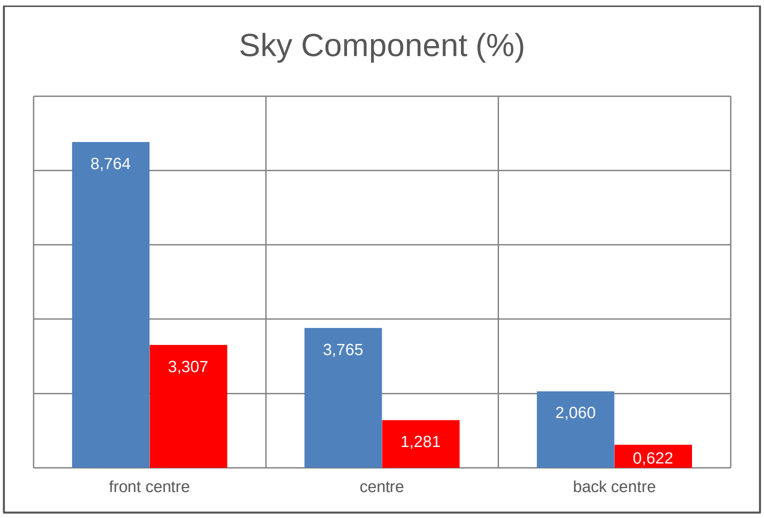
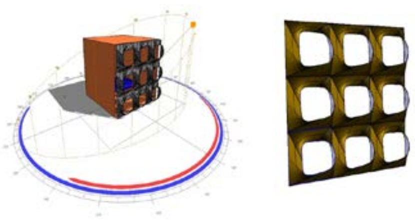


Fig. 54 – Sky Component over centre nodes on grid SD #1 (blue) and SD #2 (red).

Direct solar gains are a critical contribution to effective passive solar design. Solar radiation does not only heat up the fabric of the building, but enters the living space through the glazing and will directly warm up the space inside. This can be much wanted at times, but it will also be necessary to avoid peak temperatures on other periods. Adequate shading devices allow for maximum heat gain in the winter months and need to protect from unwanted heat in the summer months.

The same cube with the same shading devices as for the calculation of the sky component was used to perform a thermal analysis and obtain the exact contribution of Direct Solar Gains to the thermal performance of the test cube (Table 1).

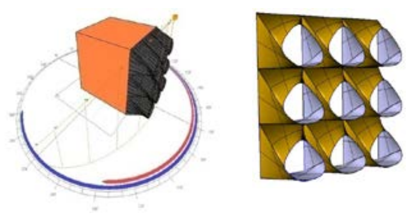
DSG Thermal Analysis #1



ANNUAL LOADS TABLE
Direct Solar Gains - Qg
Cube TEST 09122010 - Monthly Averages

HOUR	JAN (Wh)	JUN (Wh)	SEP (Wh)	DEC (Wh)
-----	-----	-----	-----	-----
11	899	744	1599	1090
12	1118	821	1855	**1241**
13	1286	**878**	1796	1527
14	1339	862	1541	778

DSG Thermal Analysis #2



ANNUAL LOADS TABLE
Direct Solar Gains - Qg
Cube TEST 09122010 - Monthly Averages

HOUR	JAN (Wh)	JUN (Wh)	SEP (Wh)	DEC (Wh)
-----	-----	-----	-----	-----
11	866	677	1538	1063
12	1079	738	1784	**1201**
13	1241	**791**	1716	1475
14	1300	760	1458	728

Table 1 – Sample of Direct Solar Gains Analysis for Shading Device #1 (left) and #2 (right).

As mentioned in the description of the experimental set up, the shading devices are mounted at a distance of 100 mm in front of the cube with the intent to simplify and speed up the optimisation routine by avoiding time consuming calculations of inter-zonal dependencies. So, in order to get a quick verification of possible interference in the results of the small distance of 100mm at which the shading devices are placed in front of the cube, the same analysis procedure was rerun but this time with the shading device #2 fixed to the cube. No measurable difference was found, as can be observed in the results (Table 2), comparing the columns SD #2 and SD #2 attached. Since the calculation of inter-zonal dependencies is time consuming, but are irrelevant for the analysis and simulation of shading devices on only one enclosed space, based on the results it was thus concluded that for simulation purposes, shading devices can be placed in front of the cube at a distance of 100mm without any influence on the final results.

Sky Component in % nodes	SD #1	SD #1 blocked	no SD	SD #2	SD #2 atached
front left	8,34	7,03	23,81	3,39	3,39
front middle	8,76	3,74	26,25	3,31	3,31
front right	8,72	3,16	23,68	3,14	3,14
midle left	4,31	2,71	11,33	1,70	1,70
midle midle	3,77	1,30	11,63	1,28	1,28
midel right	3,53	1,15	10,45	1,44	1,44
back left	1,83	1,15	5,19	0,71	0,71
back middle	2,06	0,73	6,05	0,62	0,62
back right	1,71	0,25	5,33	0,34	0,34
average	4,78	2,36	13,75	1,77	1,77

Table 2 – Sky Component in % for the different Shading Devices.

In this first part of Case Study 2, the objective of digital simulation was to study the influence of different shading devices on natural lighting and on direct solar gains. Preliminary analyses showed that the values for Natural Lighting calculated over an analysis grid with 9 nodes are clearly different and point, as expected, to small differences between shading devices. Thus optimisation towards this objective seems possible and probably even more constraining and different requirements can be applied to the optimisation procedure. It also seems plausible to link this quantitative optimisation objective to the personal aesthetic perception of the designer, because the parametric construction of the shading device allows for a combination of random variables which result in distinctive formal outcomes.

Direct solar gain, a very important component in passive solar design, does not vary to the amount as was expected at first, but still small and coherent differences can be measured between different shading devices, with better visualisation of those differences if all other factors that influence direct gains are excluded in the calculation. Thus it was shown that optimisation towards this objective can also be accomplished (Fig. 55)

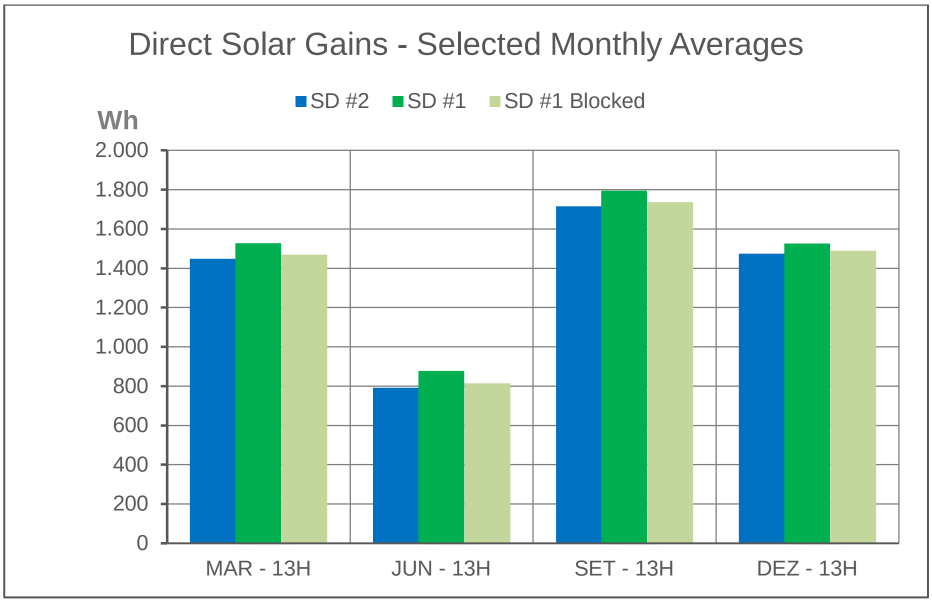


Fig. 55 – Direct Solar Gains on four different moments in time for the different Shading Devices.

4.2.3 Optimising Shading Devices

Having verified the suitability of all the items required for an integrated optimisation, the same virtual configuration of the test cube was built as described in the paragraph 4.2.1, again with the objective to be as similar in dimensions and characteristics as the test cell built on the campus of the University of Minho. The characteristics of the materials for the walls, the floor and the roof were changed to match the characteristics of those materials used on the test cell. The south facing facade is almost entirely closed by a single sheet of transparent PVC film (2200x2200mm) simulating a real single pane glass window. However some of the properties of this PVC film which do not have any influence on the thermal analysis of the model, were set to match transparent glass allowing for better representation of transparency in the interface of Ecotect. Except for this south facing facade, all the other faces of the cube, including the roof and the floor are made out of 80mm thick Polystyrene Foam panelling Dow Roofmate SL-a (Dow, 2011). All those walls are covered on the outside with a white opaque PVC film for protection and for minimising eventual direct gains of the fabric of the cube.

The shading device was then optimised for four objectives, being one the aesthetical perception of the decision maker (DM - Designer) and the other three being the variables for simulation and analysis:

- Direct Solar Gains, in Wh on Julian Day 172 (21/6), in order to minimise the increase in temperature inside the space by blocking the sunlight which enters through the window. On this day the sun is at the highest point at the azimuth and blocking direct sunlight would result in the minimal depth of the shading device. This value needs to be minimised (Fig. 56 – left).

- Direct Solar Gains in Wh on Julian Day 355 (21/12), with the intent to maximise the increase in temperature inside a space by not blocking the sunlight which enters through the window. This value needs to be maximised (Fig. 56 – right).

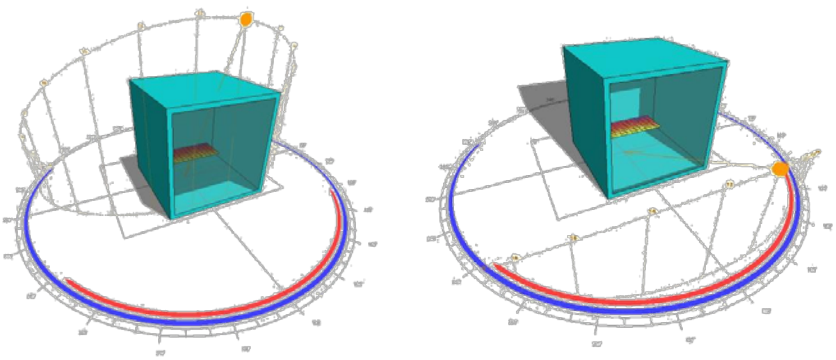


Fig. 56 – Solar Path Diagram on different Julian Days.

- Average Daylight Factor between 3 points on an analysis grid in order to maximise the daylight quality inside a room. This value needs to be maximised (Fig. 57).

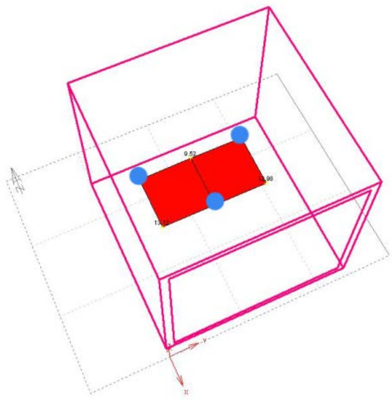


Fig. 57 – Average Daylight Factor between 3 points on the analyses grid.

Again, and as was also applied and demonstrated in the previous case study, optimisation towards the combination of both quantitative and qualitative evaluation criteria of the design's performance will demonstrate the effectiveness of the proposed methodology and the software construct built as the supporting tool.

The MOEA adopted in this work is the Reduced Pareto Set Genetic Algorithm (RPSGA) proposed before by Gaspar-Cunha and Covas (2004) and was executed in close collaboration with the developer of this particular Evolutionary Algorithm, which was also successfully tested in the first case study. The values of the parameters inside the RPSGA are the best-practice values as described in his research.

Interactive optimisation towards the aesthetical preferences of the designer in this case study was possible combining iteratively "a priori" and "a posterior" methods, as described in paragraph 4.1.3 in the previous case study.

An initial population of one hundred individuals (Fig. 58) was created and a roulette wheel selection strategy was adopted. In all of the following runs, the optimisation algorithm was set to run for sixteen generations, in order to limit the computation time required by the analysis and simulation software.

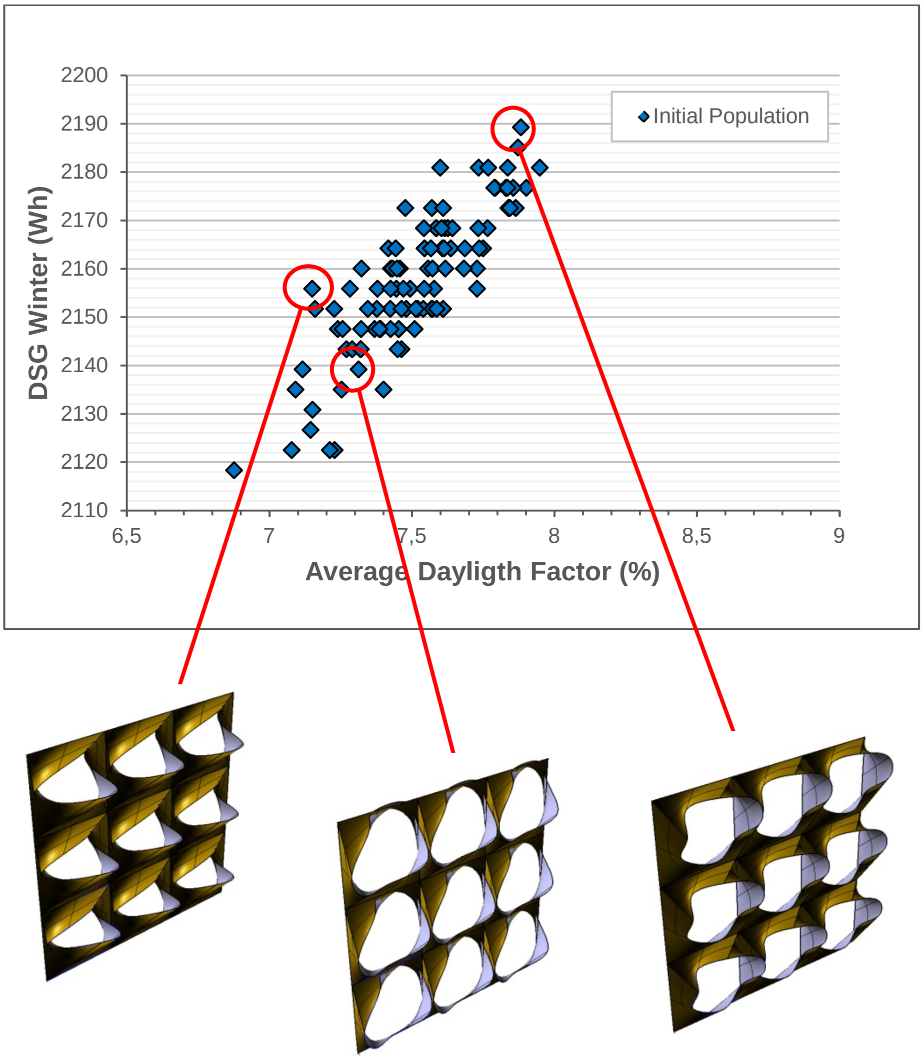


Fig. 58 – The initial population of one hundred individuals.

Optimisation Run for DSG Summer

In the first optimisation run, the optimisation algorithm was used without the interference of the decision maker (Designer). The variables and the objectives for optimisation were determined as described in the previous paragraphs, and the range of variation of the variables was decided upon, and introduced in the optimisation algorithm. In this first optimisation run, the shading device was optimised for only two different variables: Direct Solar Gains in summer time, has to be as low as possible, and the Daylight Factor, which value must be as high as possible. No specific weighting factors were attributed to the different variables. This is the simplest situation but qualifies as an important first step in an interactive optimisation process. The results will contribute with valuable information about the possible solution space and, in particular, feedback will be provided about the differentiation of possible solutions that can be expected. Visual representations of some of the solutions (Fig. 59) illustrate what alternatives populate the solution space, and how optimisation towards the introduced variables and their range of variation changes and transforms the form and geometry towards the objectives imposed upon the optimisation algorithm.

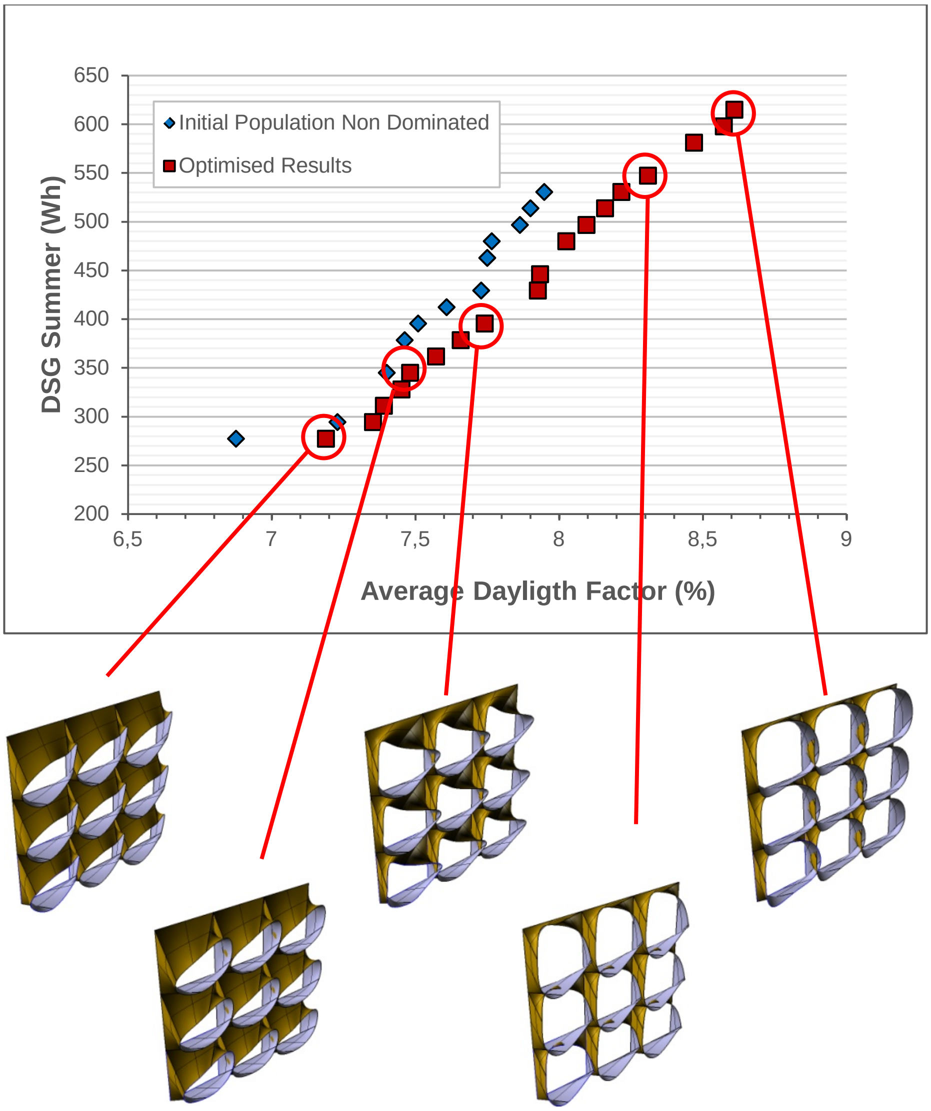


Fig. 59 – First optimisation run without specific weighting factor and visual representations of some of the solutions.

The graph and the three dimensional representation also illustrates the expected trade-off between the different objectives which the designer will have to make opting for one solution or another

Analysing the graph one can clearly notice how the Pareto frontier of optimised results (in red in Fig. 59) is a different set of solutions than the set of non-dominated individuals of the initial population and with an evident tendency for better response to the main objectives for optimisation: low value for DSG in summer and high overall Average Daylight Factor (DLF).

When the optimisation algorithm is run within a graphical user interface, three dimensional representations of some of the solutions (and eventually also some of the elements of the initial population) can provide the designer with a quick and simple visual verification of the expected geometrical *behaviour* of the shading devices and a general perception of the (geometrical) solution to expect.

Optimisation Run for DSG Winter

Subsequently, for a second optimisation run the optimisation algorithm was used again without the interference of the designer and no weighting factors were applied. All variables, the range of variation of the variables and the objectives for optimisation were exactly the same as in the previous optimisation run. For this optimisation run, the shading device was again optimised for only two different variables and without attribution of any specific weighting factors: this time Direct Solar Gains in winter time has to be as high as possible, and again the Daylight Factor which value must be as high as possible. It was expected that also these results will contribute with useful information about the further development of the design process.

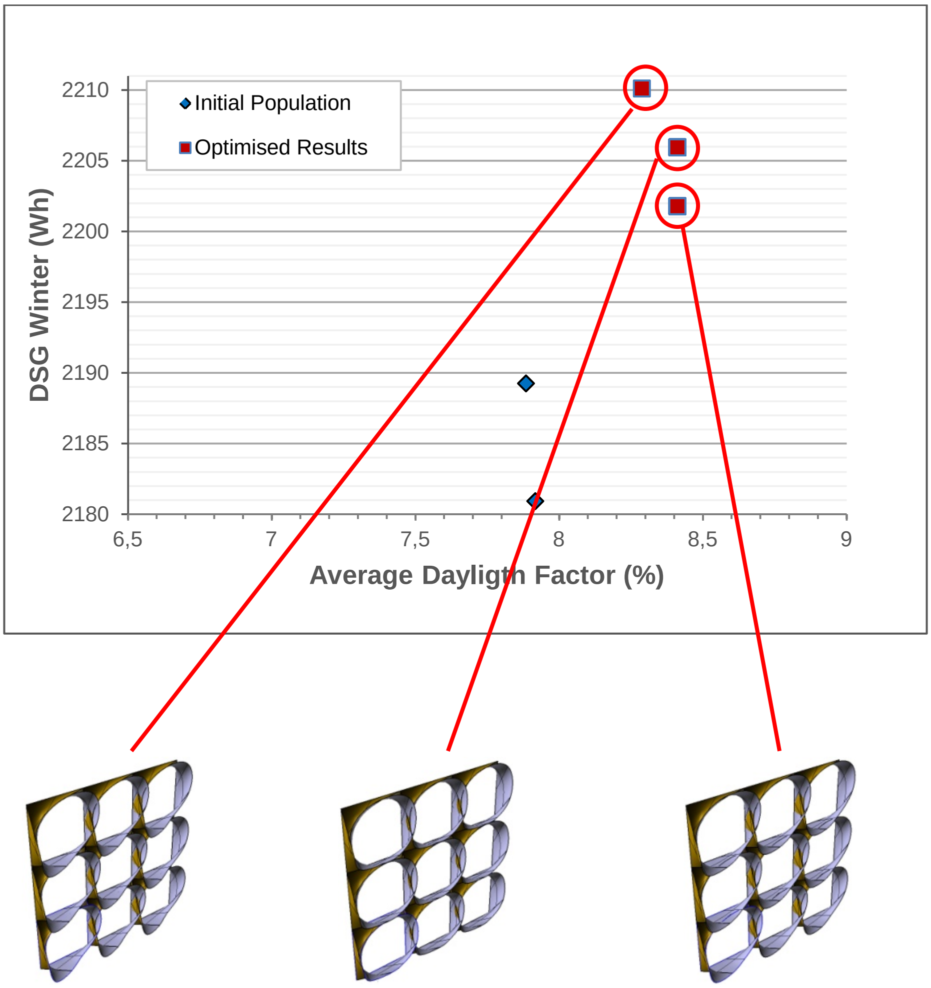


Fig. 60 – Second optimisation run without weighting factors depicting visual representations of some of the solutions.

The conclusions after analysing the graph are similar to the conclusions after the previous optimisation run. It can be observed again how the Pareto frontier of optimised results (albeit a very reduced set of apparently very similar solutions - in red in Fig. 60) have quite a different behaviour than the set of non-dominated individuals of the initial population. Visual observation of some of the solutions confirms empirically what can be expected of an optimised solution for winter time, where a shading device has to contribute to improve DSG and to better Day Lighting.

Multi-Objective Optimisation

In this optimisation run the concept multi-objective optimisation with three different variables was introduced. The objectives in this run are Direct Solar Gains in summer time, which have to be minimised. Direct Solar Gains in winter time, which have to be maximised, and the Daylight Factor which is a measure of "openness" of the structure, and hence does also inform, although indirectly, about the lightness of the shading device.

Originally it was expected that no dependency would exist between the Direct Solar Gains (DSG) in winter time and the DSG in summer time (see also Fig. 61 and Fig. 62).

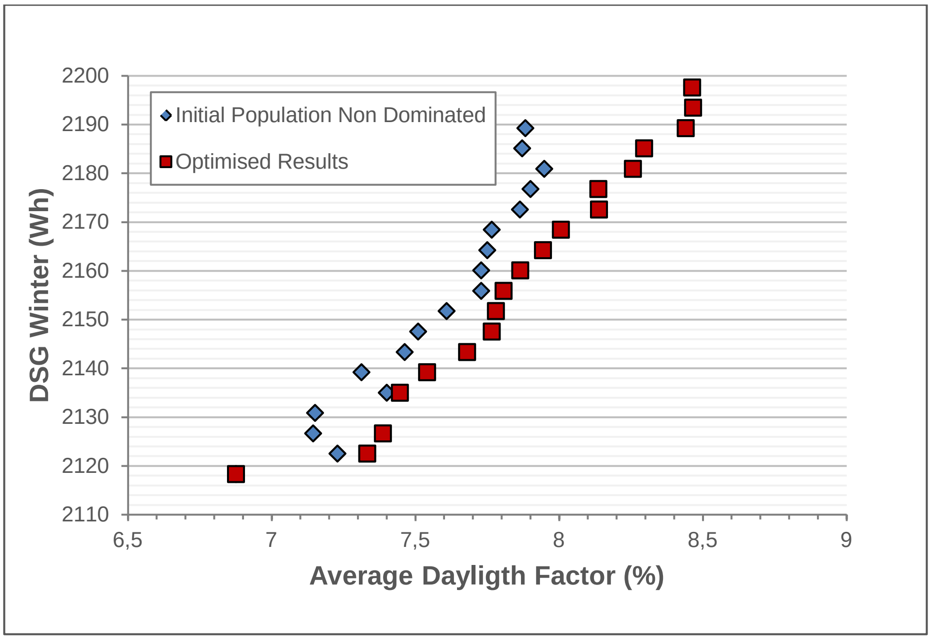


Fig. 61 – DSG Winter and Average Daylight Factor: Non Dominated members of the initial population and Pareto Optimum results after the first optimisation run without attribution of weighting factors.

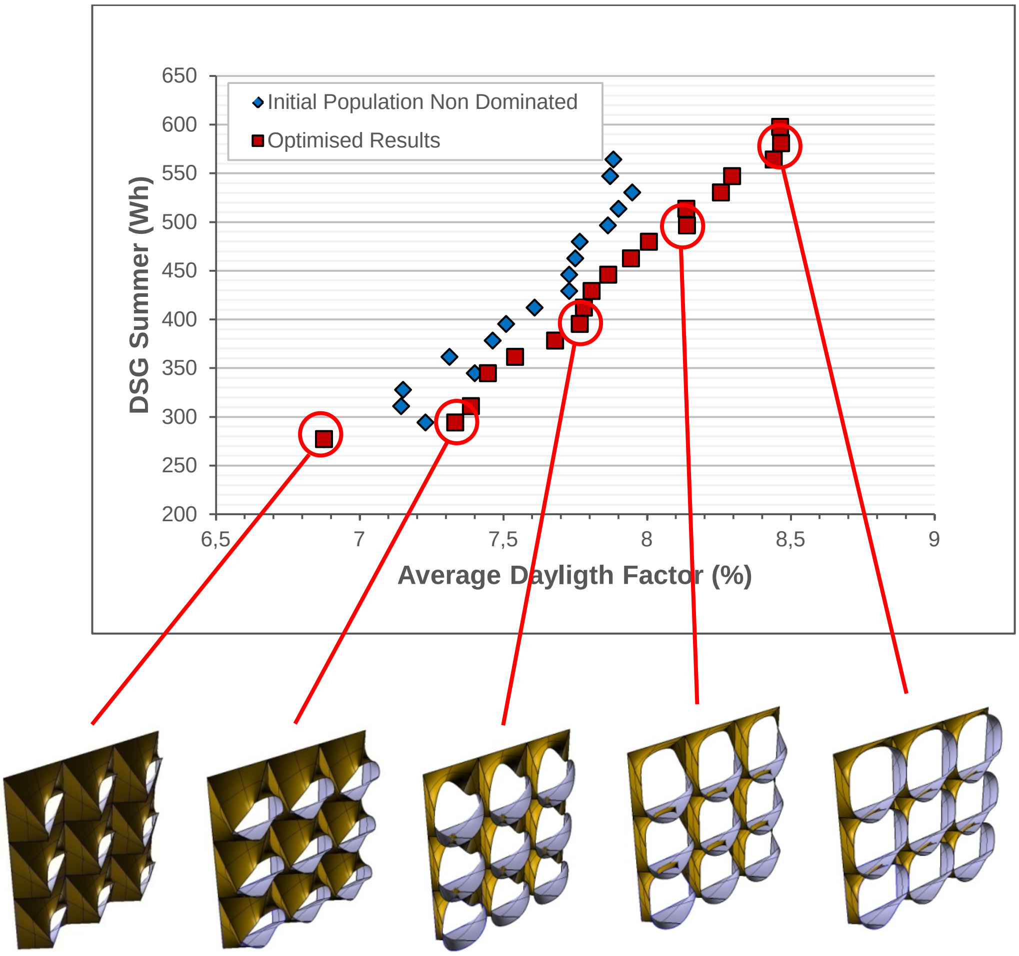


Fig. 62 – DSG Summer and Average Daylight Factor: Non Dominated members of the initial population and Pareto Optimum results after the first optimisation run without attribution of weighting factors and with illustration of some of the solutions.

However, closer analyses of the results show a linear dependency between the two objectives for optimisation, as depicted in figure 63. Though unexpected, this has no effect on the evaluation of the results, which is clearly visible by comparing figure 61e figure 62, showing exactly the same graphical configuration. Anyway, by changing the design, both parameters are affected and the designer is challenged to find the right balance between the two, e.g. by optimising the overall comfort hours.

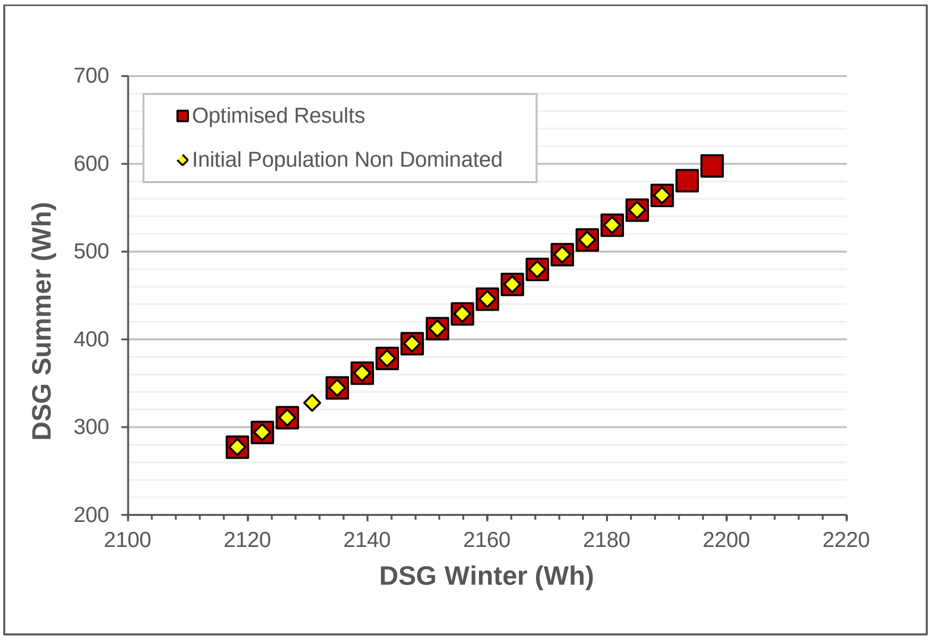


Fig. 63 – Pareto Optimum results after the first optimisation run without attribution of a weighting factor – depicting linear relation between DSG Summer and DSG Winter.

The linear dependency between two variables is an example of an occurrence which was not planned and which intuitively could not have been foreseen at the beginning of the optimisation process. It demonstrates how information gathered during the optimisation process is at the basis of interaction between the designer and the digital medium. If, in the former example, linear dependency between the variables would severely reduce the solution space to geometrically very similar solutions, another completely new iteration cycle can be decided upon with a different parametric starting geometry or with optimisation to different variables and alternative goals. This underlines the main objective and the intended goal of digital design optimisation, which – as stated in chapter 2 - is precisely supporting decision making for those problems where an intuitive approach based on experience and empiric knowledge do not offer any guarantee on expected outcome and thus can no longer support the (digital) design process.

Also, and as an immediate consequence, in this particular case it is clear that the optimisation does not have to produce any 3- dimensional Pareto frontier (surface) and that the combination of two different 2-dimensional representations do provide sufficient information about the optimised solutions.

Having established this insight the geometry or geometries selected by the designer are now used as the next initial population for the final optimisation cycle together with a weighted significance of the optimisation objectives which is also introduced in the algorithm.

An extra multi-objective optimisation run was executed with some of the solutions located on the Pareto frontier of the previous run. The aesthetically most interesting were selected and were reintroduced in the optimisation software. However, the new Pareto frontier which resulted of this operation lies very close to the Pareto frontier from the previous run and the newly optimised solutions are not that diverse from the previous run. The Pareto frontier was slightly more concentrated and confined to a specific part of the previous Pareto frontier, but the differences with the previous optimisation run were considered as insignificant.

Multi-Objective Optimisation Run with induced relative importance of the variables

The ultimate ambition of the optimisation process in this case study is to build a shading device which guarantees better comfort all year round. Comfort can be the main driver for decision for further development and should be based on the analysis of the Total Comfort Degree Hours. An example is calculated for the test cube without any shading device in Guimarães (Fig. 41). By determining a Comfort Zone between 26ºC and 18ºC the thermal analysis software from Autodesk Ecotect can provide a graph showing the total hours outside the Comfort Zone, either too cold or too hot. Analysing the graph (Fig. 64) it is clear that the temperature inside the cube in winter time is in a considerable period of time below the minimum comfort temperature, and that on the other hand the temperature in summer time does exceed the maximum comfort limit for less hours.

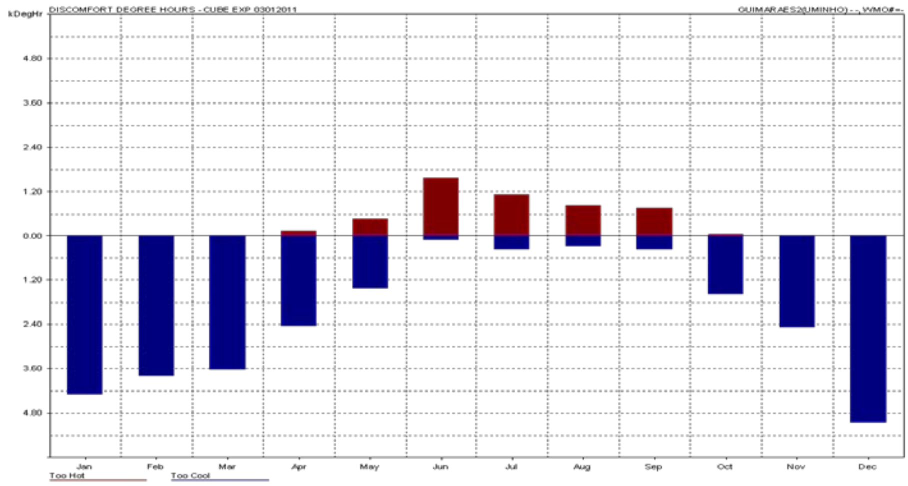


Fig. 64 – Total Comfort Degree Hours for the test cube in Guimarães in 2003 without any shading device.

A more detailed study of the results of the comfort hours analysis of the present test cube - which was not specifically built with the aim to evaluate shading devices through comfort hours –showed that in this specific case (due to geometrical constraints of the test cube) the Direct Solar Gains contribute in only a very small amount to the overall total comfort hours calculation, thus inhibiting the use of this variable as a discriminating indicator of shading device performance. Thus it was decided in this case to use Direct Solar Gains directly, rather than Total Comfort Degree Hours, to execute optimisation runs on shading devices, with the introduction of a weighted significance in order to tailor overall performance.

Based on the importance for Direct Solar Gains in winter time this variable was attributed a 0.5 importance factor and the variable for Direct Solar Gains in summer time was attributed with a 0.2 importance factor. Good natural lighting is always important, summer and winter, and was attributed a 0.3 importance factor. If for some reason direct sunlight entering the space would disturb the quality of living inside, by exaggerating glare for example, the installation of a movable interior shading device, such as a screen or a curtain, should be highly recommended.

As can be seen in Fig. 65, the introduction of different weighting factors for the optimisation variables clearly *pushed* the Pareto frontier of optimised solutions to the upper right corner. This is precisely as could be expected for the increased importance of the DSG Winter variable

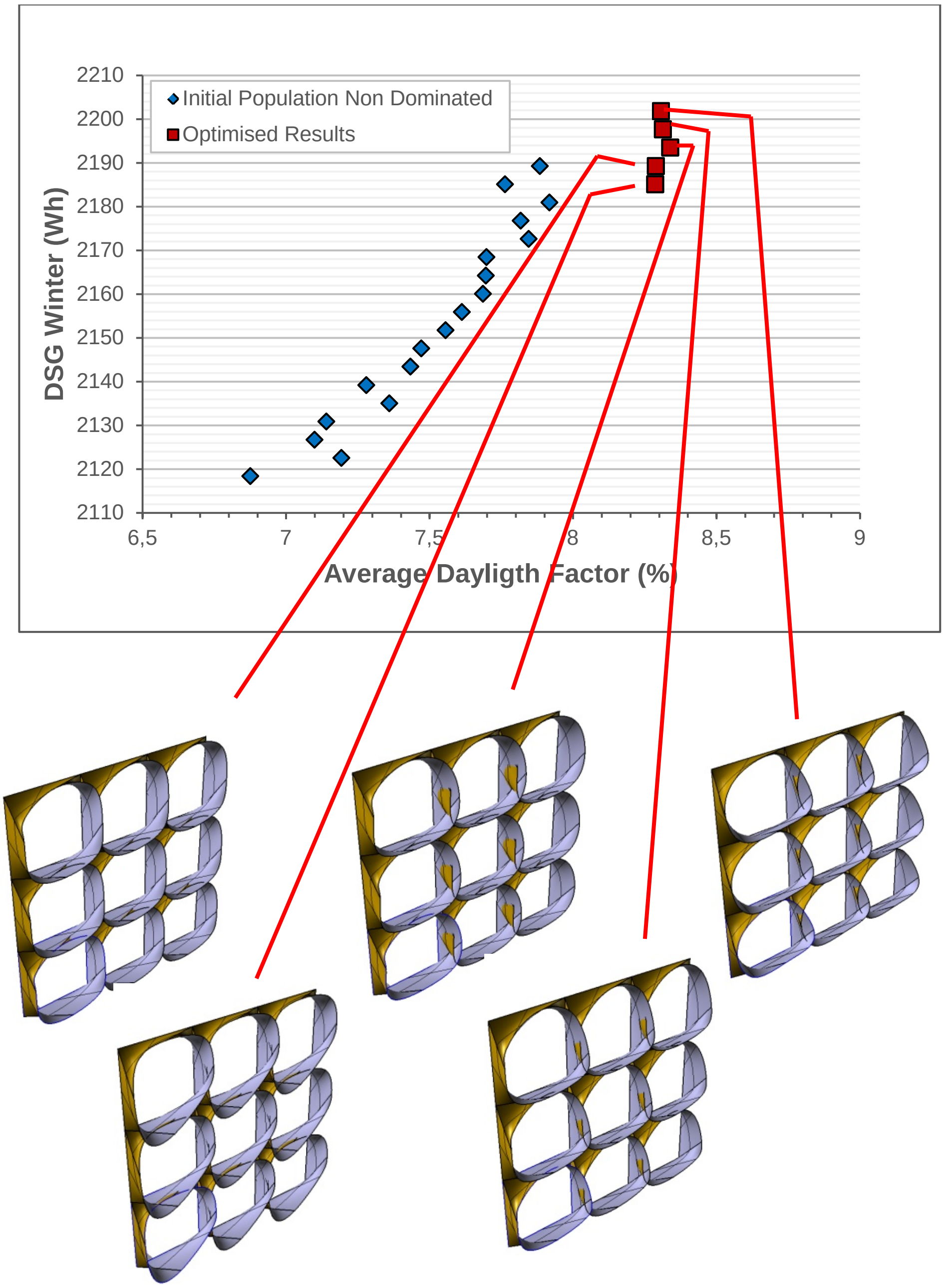


Fig. 65 – Final optimisation results after introduction of Weighting Factors.

Furthermore, the weighting factor of only 0.2 for the DSG Summer variable compared to the weighting factor of 0.5 for the DSG Winter variable results in *shorter* and more *open* shading devices, as is clearly seen in the examples in Fig. 65.It is also observed that the geometrical results of this optimisation are very similar, although resulting in differing actual values of the DSG Winter variable.

In addition, a sixth (demonstrative) optimisation run was performed, with different weighting factors, not only to demonstrate how weighting factors influence the optimisation process but also to demonstrate the outcome of a different set of optimised solutions. In this run the following weighting factors were applied, representing almost the opposite philosophy of the previous run: Daylighting 0.1, DSG Summer 0.6 and DSG Winter 0.3. The optimisation results are shown in Fig. 66.

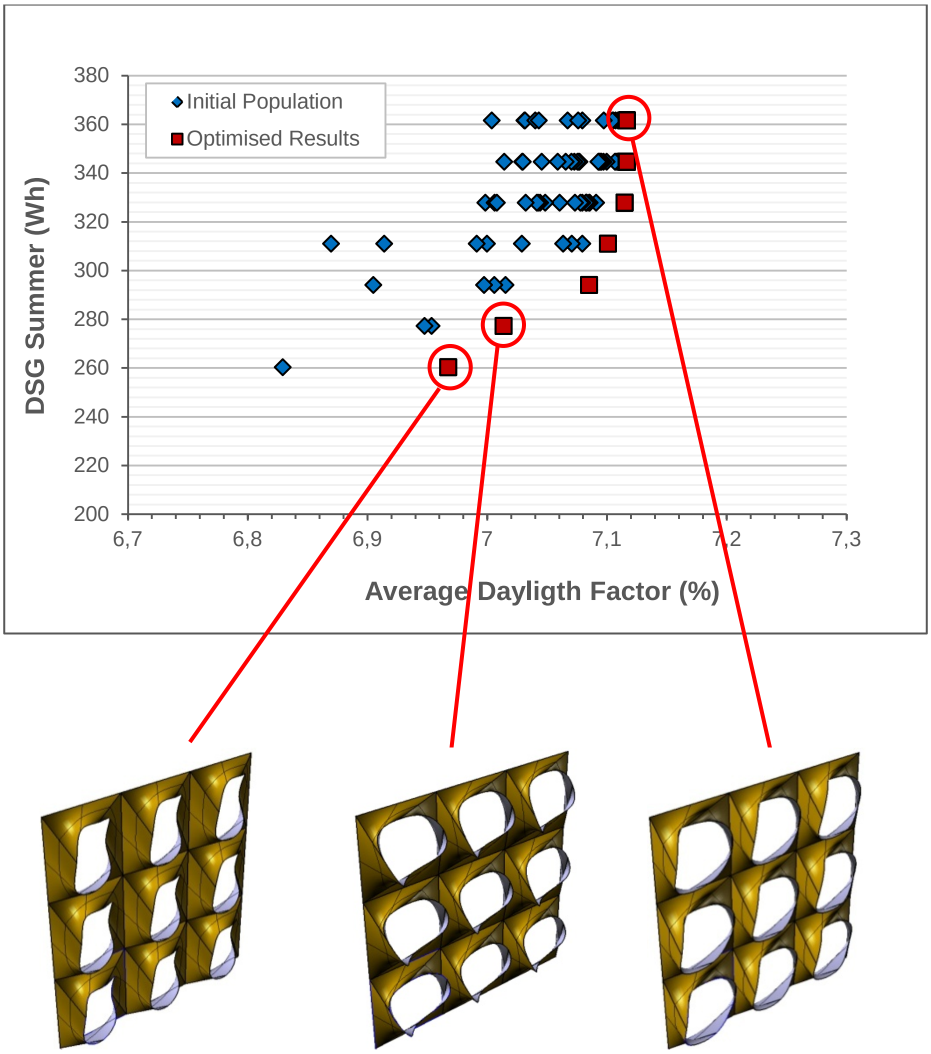


Fig. 66 – Final optimisation results after introduction Weighting Factors.

The results from this optimisation run (Fig. 66) clearly show differences with those from the previous run (Fig. 65). As different Weighting Factors were introduced in the optimisation algorithm, different solutions can be observed along the Pareto frontier. By attributing a weighting factor of only 0.1 to the variable of Average Daylight Factor this objective was reduced to least important objective for optimisation. This becomes quite obvious by observing the representations depicted in Fig. 65 with those depicted in Fig. 66, were this particular variable had a weighting factor of 0.3. The shading devices where daylight was weighted less, clearly show a more closed geometry (Fig. 66) as compared to those where daylight was weighted more (Fig. 65)

In Figure 67, the Direct Solar Gains in wintertime (DSG Winter) and summertime (DSG Summer) are depicted for the three different situation, described in this paragraph: without shading device and shading devices with different weighting factors. From the picture the resulting differences in Solar Gains are clearly visible. Where without any shading device both in summertime as well as wintertime, DSG are maximal, the use of shading devices with 'tailored' (through weighting factors) shading behaviour, clearly result in more DSG in summer- and winter time respectively.

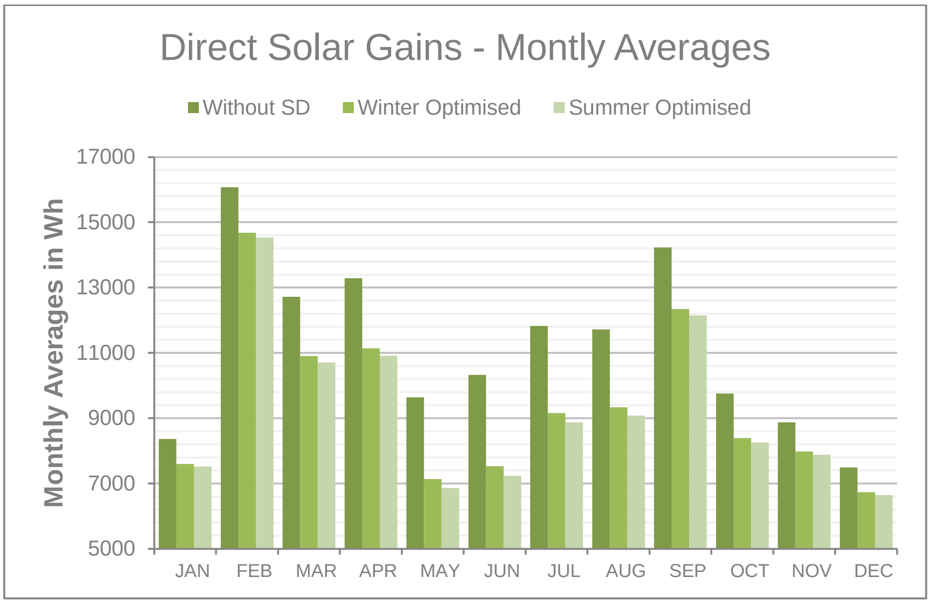


Fig. 67 – DSG – Monthly Averages for two different SD (Fig. 66 left – Fig. 65 right) from different regions of the combined Pareto frontier compared to a simulation without any SD.

By comparing all the results in only one graph (Fig. 68) it becomes clear that interaction can be achieved by means of introducing weighting factors according to the importance of the variables. While an optimisation run of a similar parametric geometry with similar optimisation but without the application of any weighting factors will produce a broad Pareto Optimum frontier (see Fig. 61 e Fig. 62 from the Multi-Objective Optimisation), the introduction of a weighting factor, will concentrate on a reduced solution space and show optimised solutions along a reduced Pareto frontier in a specific area. This becomes quite clear by comparing the Pareto Optimum frontier of Run 5 with Pareto Optimum frontier of Run 6, both representing extreme cases of the non-weighted Run 3.

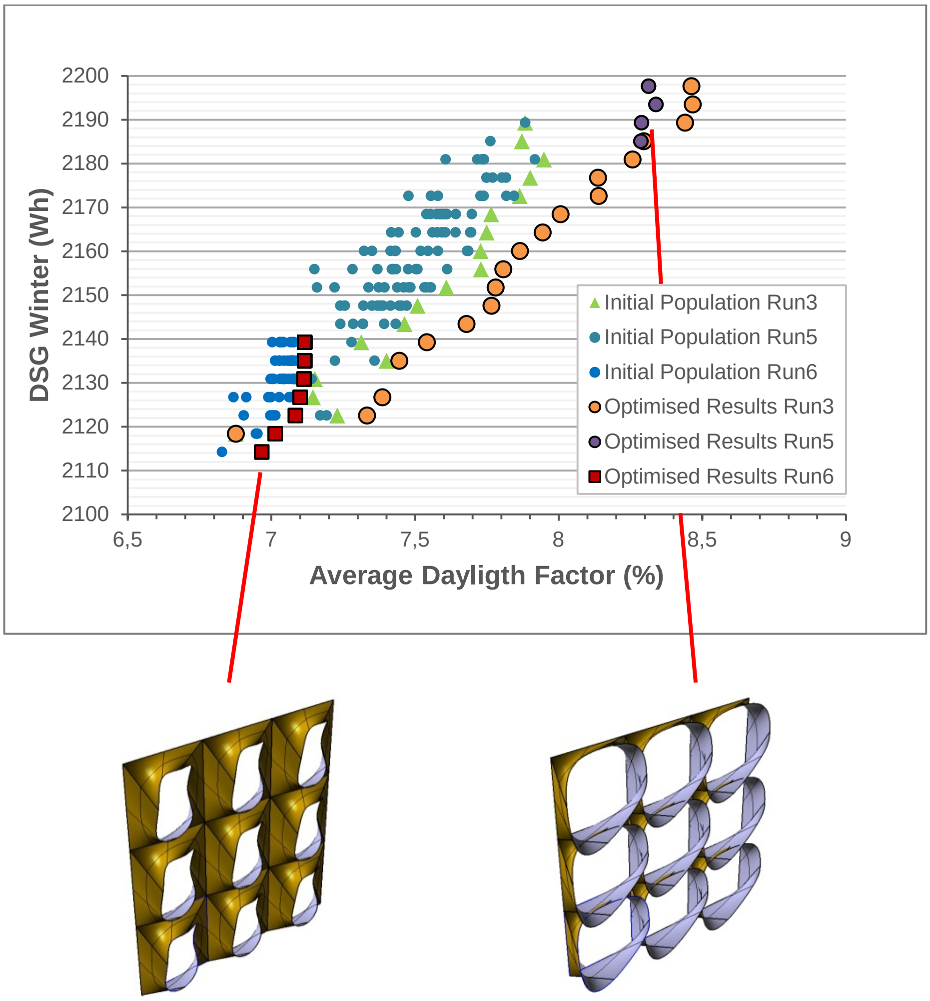


Fig. 68 – Comparing Multi-Objective Optimisation with induced relative importance of variables.

Two representations of a shading device belonging to either of the differently optimised solutions are presented in figure 68. It is the designer's decision to opt for one or another solution for further development in the detailing phase of the design project, if reducing the direct solar gains in summer time is more important, weighting factors according those objectives can be introduced in the optimisation algorithm, if on the contrary the designer expect more benefits from direct solar gains in wintertime, the solutions along the Pareto frontier of Run 5 are all suitable for this kind of design strategy. The complete set of the first set of non-dominant optimised solutions from the multi-objective optimisation with induced relative importance of variables are shown in Figure 69, and some examples of the set of Pareto Optimum solution from the second set of non-dominant optimised solutions from the multi-objective optimisation with induced relative importance of variables 6 are shown in Figure 70. The non-dominant solutions for the first multi-objective optimisation are a rather small set of only 10 geometries, of which some, for no particular reason are completely identical. This does not, however, reduce the practical use of the results as a start for further development.

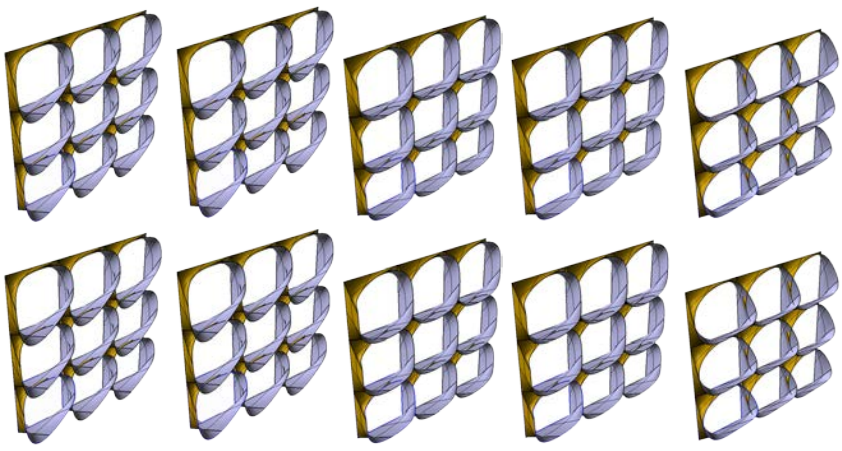


Fig. 69 –Complete set of 10 Pareto Optimum Solutions for Run5.

The full set of 68 Pareto Optimum solutions from the second multi-objective optimisation does also contain some solutions which are in part or completely identical, but overall distinctive differences can be observed between shading devices according to their position on the Pareto Frontier (see also Annex C for the full set).

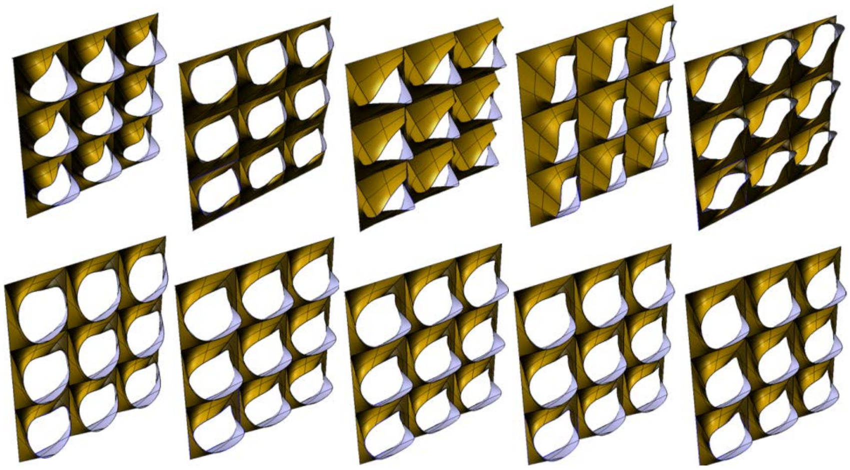


Fig.70 – Some examples from the set of 68 Pareto Optimum Solutions for the second multi-objective optimisation showing different and similar geometries.

4.2.4 Simplifying simulation

Designers and architects do not generally work on super computers, and although the computers used in the case-studies presented in this thesis were technologically up to date and with the regular hardware configurations and computational capacities, obtaining optimised results uses considerable computation time. Simulation and analysis becomes especially more time consuming with increasing number of elements of the meshes which make out the geometrical description of the object in the CAD environment.

The goal of this part of the case study is to verify if the variables, which could be used for a study in the optimisation of the geometry of shading devices, provide sufficient and detailed feedback for simulation in passive solar design. At the same time, the influence of scaling of the shading devices will be assessed by comparing the results of only one big shading module with grids of four, nine and sixteen proportionally smaller shading modules. If the results of one module are similar to the results of a grid of four, nine and sixteen modules, one module can be used for the thermal analyses which will considerably reduce the computing time since the amount of calculations is directly related to the quantity of meshes which make up the design of the shading device.

In addition lighting calculations inside the cube were also simplified and instead of using a two dimensional measuring grid parallel to the floor plan along the XY-axes, a point object was created exactly in the centre of the cube for measuring Sky Component and Daylight Factor.

Two different shading devices were used for this simulation, and just as in the previous related experiments, significant differences were expected between the simulation results of both shading devices. The shading devices were selected on expected better behaviour in winter (Fig. 71) and expected better behaviour in summer (Fig. 72).

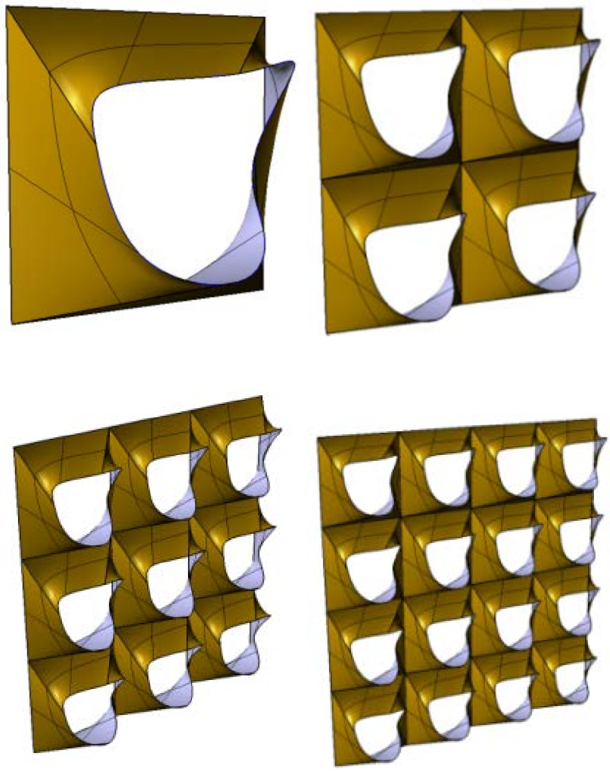


Fig. 71 –Different grid configurations for Shading Device #1. .

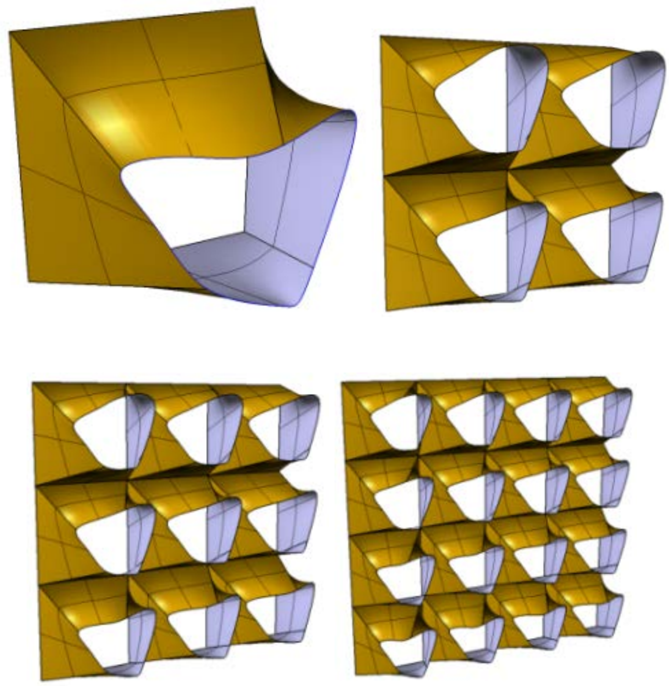


Fig. 72 –Different grid configurations for Shading Device #2. .

For the calculation of the Daylight Factor, the parameter settings of the analysis were all set for obtaining very precise and accurate results to be able to evaluate the performance of different shading devices where only very small differences in the final results could be expected.

The total area of the resulting two dimensional figure of the projected curve at the front of the shading device element was also calculated in order to control the "openness" of the shading device. This is an important feature of any shading device. A less obstructive shading device, which allows for more visibility from the inside to the outside, is desirable.

The thermal gains and losses were calculated for the winter months, from December 1st to April 30th. The Portuguese Thermal Regulation (DL nº 80/2006 from 4/4/2006) defines the heating season with the duration of seven months, but the weather file of Guimarães used in this study and compiled locally on the site of the campus of the Minho University during a previous research project illustrates that in this particular location, a heating period of only five months can be considered (Fig. 73)

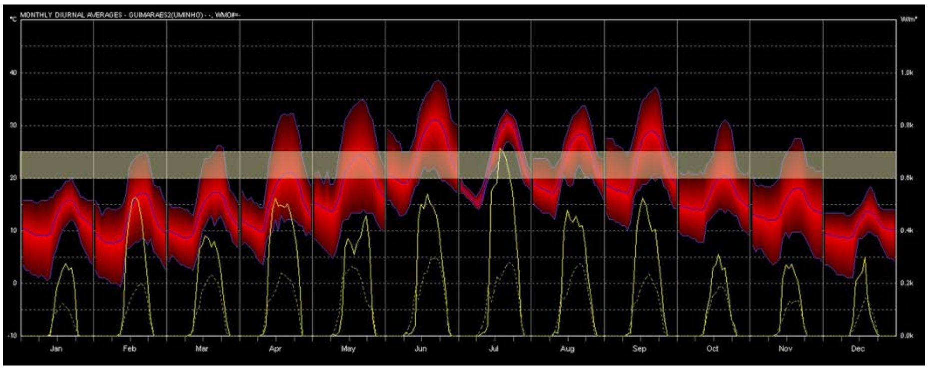


Fig. 73 – Weather file of Guimarães with depicted comfort zone between 20ºC and 25ºC and a 5 month heating season.

The relative contribution of Direct Solar Gains to the total amount of thermal gains in this period was used as the reference value as an indicative of to the efficiency of the shading device.

Shading Device #1	1 Module	4 Modules	9 Modules
Area at the front in mm2	1836197	1836196	1836198
Sky Component in %	1,15421	1,6105	1,15421
Daylight Factor in %	6,23316	7,01457	6,23316
Direct Solar Gains in %	33,3	33,3	33,3

Shading Device #2	1 Module	4 Modules	9 Modules
Area at the front in mm2	1881433	1881480	1881477
Sky Component in %	0,825336	0,885512	0,881673
Daylight Factor in %	6,12425	6,5224	6,3516
Direct Solar Gains in %	33,2	33,2	33,2

Table 3 – Comparative Results of the calculations for one, four or nine scaled modules of the same shading device.

4.2.5 Simplifying simulation: conclusions

As stated above, running performative simulation software with an object composed out of less meshes will reduce the calculation time significantly. Some quick experiments with the calculation of daylight factors with an identical shading device module but scaled to fit in a grid with either only one or with four, nine or sixteen modules, did not result in different values of the daylight factor in the centre of the cube at a height of 700mm (Table 3). Therefore, all the calculations can be done over only one module, resulting in reduced computation time and faster results. Then, afterwards, one can present the optimised solutions for visual selection in grids of nine, sixteen or more modules.

4.2.6 Analysing Shading Devices on a Generic Dwelling

In the previous section was shown that an integrated optimisation method can be used to design optimised shading devices that outperform conventional alternatives. However, all was done on a test cube of dimensions not realistic for actual living spaces.

To demonstrate and compare the qualities of an optimised shading device of the kind developed in this study, digital simulations and calculations were performed on a generic one-bedroom house with an optimised shading device and compared to a standard configuration with normal windows, a randomly chosen shading device and an optimised shading device but scaled down to half its size. As a reference the calculations were also performed on a set up without any shading device (Fig. 74 e Fig. 75).

In these simulations Day Lighting was calculated on a point at 700mm height at the centre of the living room and at three quarters of the depth of the sleeping room, as prescribed in the European building specifications (Santos, 2001).

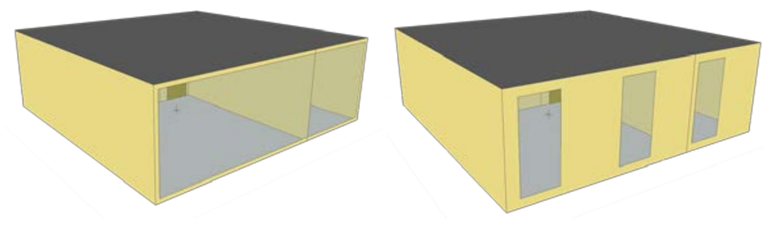


Fig. 74 – Generic House with standard windows (a), without any shading device (b), with a non-optimised shading device (c), with an optimised shading device (d) and with an optimised shading device 50% scaled

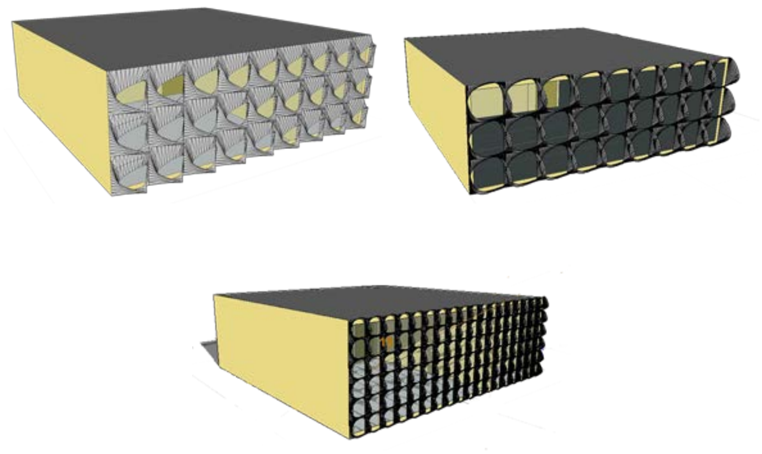


Fig. 74 (cont.) – Generic House with standard windows (a), without any shading device (b), with a non-optimised shading device (c), with an optimised shading device (d) and with an optimised shading device 50% scaled

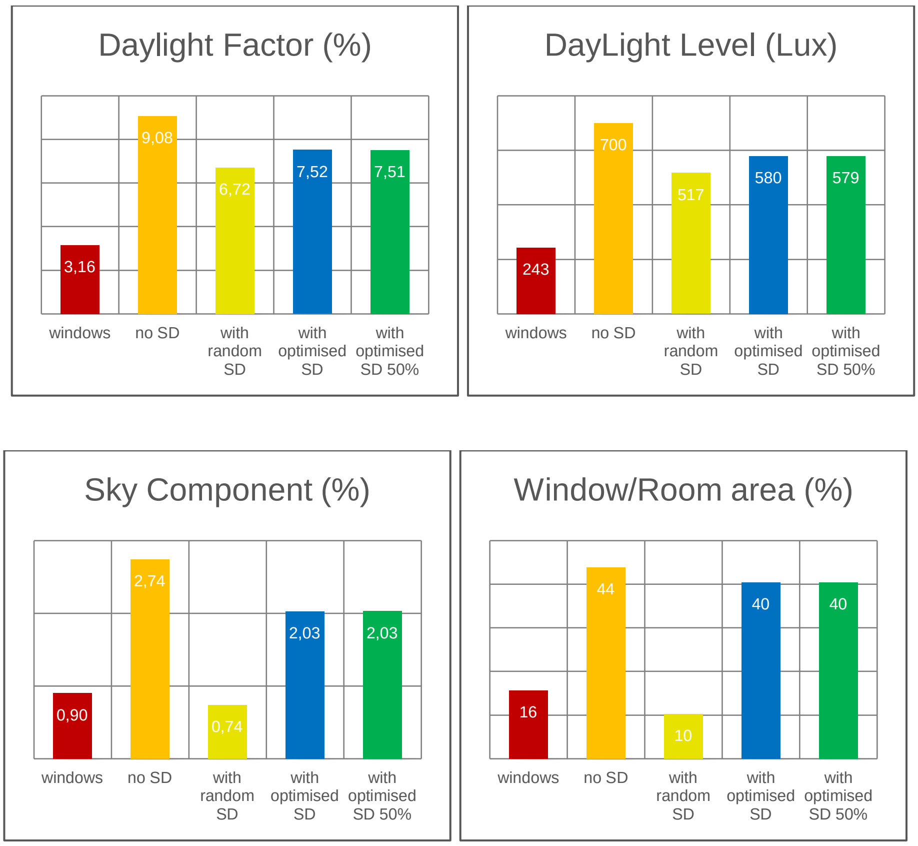


Fig. 75 – Comparing results with the application of the SD on a generic house.

One of the basic assumptions supporting the concept of facade shading is the construction of a grid of shading devices. The exact number of components and their size should be a perceptual personal option of the designer. This has been demonstrated in the simulation with the application of an optimised shading device on a generic housing unit, simulation results were similar between a grid of shading device units and a grid of the shading device units with identical geometry on a 50% scale (Fig. 75).

The results of this demonstration show that the application of an optimised shading device in any scaled dimension of its basic optimised geometry provides better results than a non-optimised shading device with similar geometry. Furthermore, comparative analyses of the calculated data demonstrate that a shading device can perform better on all parameters involving lighting than standard fenestration. But also, as was demonstrated in the previous paragraph, no difference can be observed between identical shading devices sized to a different scale. Only if the wall has been removed entirely, which is of course can only be considered as a highly hypothetical situation, lighting results are higher than with a shading device. However, if Direct Solar Gains have to be accounted for, substitution of an entire wall by full glass windows and without the application of an external shading device will render living in such a place very difficult if not completely impossible, especially in summer time.

5 Conclusions and Future Work

This thesis describes a design method where the use of evolutionary principles supports the designer to better accomplish goals and objectives by stimulating interactive exploration.

The characterisation of designing as problem solving, although no longer the dominant way of thinking of design researchers, needs to be re-integrated in the design paradigm. Introducing the methodological use of numeric evaluations in the conceptual phase of the design process implies that parts of this process are, or have to be, approached as pure problem solving. If design is no longer only a search process but also involves the reformulation of the search space, novel methods and tools have to be developed and researched, which can make a valuable contribution to the exploration of the full potential of a digital design process. This was the aim of the current work.

The proposed method proved to contribute to creative exploration in the conceptual phase of the design process. This process of enhanced ideation and exploration of a considerable quantity of possible solutions inevitably leads to the generation and construction of conceptual solutions which are much richer than what straightforward problem solving of a limited series of proposals can achieve. From this perspective the use of Genetic Algorithms does not try to avoid bad or inadequate solutions but focuses intentionally on desirable and appropriate outcomes.

This research also demonstrated that the quality of the initial parametrical model is intrinsically related to the final outcome of the optimisation process. Other research also shows that it is not at all straightforward to construct geometrical models with adequate parametric variation and creative solutions are never guaranteed as an outcome (Marin, Bignon, & Lequay, 2008)(Hernandez, 2006). Even the application of more sophisticated techniques to generate form through evolutionary computing does not always guarantee acceptable solutions. (Nishino, Takagi, Cho, & Utsumiya, 2001).

Design has to deal with wicked problems (Rittel & Webber, 1973) and is usually considered as open-ended problem solving, as such it can be expected that no *universal software construct* can be proposed that is generally applicable for all kind of design processes, even the presentation of the schematic outlines of the proposed software construct in this thesis could (and should) be subverted and completed whenever the characteristics of the design process in question asks for a tailor made optimisation approach. The construction of a satisfactory software construct completed with the proper simulation and analysing software and the development of different scripts will become an integral part of the design process and one of the tasks of the designers involved in the design process. For open-ended problems the solution depends on the designers' personal interpretation of facts and reality, on what is important and what can be ignored, and on what is expected as a result.

The model proposed here does not pretend to be a universal tool but rather a method for design exploration. That is why the software construct needs dynamic multiple iteration cycles to avoid a too narrow focus on one single kind of solution too early in the design process. This difficulty can arise because of the nature of the optimisation algorithm. The optimisation algorithm uses simulation software and the results from mathematical models for evaluation. At that instance the approach can be described as a search process based on rational problem solving where logical analysis is the sole knowledge provider, making this "search" contradicting the true nature of the design as an exploration process, as was stated in the beginning of this thesis. However, the importance of iteration as a crucial feature to a digital design process, especially when generation, simulation and optimisation are included as structural characteristics, stresses the importance of constraints, repositioning and remodelling of the software construct. Gedenryd (2008) claims that iteration is *an ill-considered added feature* which subverts the proper nature of any structured modular design process. He furthermore states that "design consists of several component functions that cannot be held apart, and that display no general ordering principle among them" (Gedenryd, 2008, p.98). As a consequence of this claim he describes the design process as consisting of a number of different functions which cannot be clearly described independently and which do not oblige to any specific ordering principals among them. This confirms the claims that the software construct described in this work can only function with optimisation principals if it is problem *specific* and if it is modelled to the *personal* design principals of the designer or the design team – as was the case in this current study.

Equally, the use of generative algorithms supporting the generation of great quantities of possible design candidates is limited to the optimisation of only a reduced set of quantitative properties. The introduction of more complex attributes, such as functional principals or the qualitative evaluation of aesthetics, would inevitably require a considerably more complex software construct and would need significantly more computational power. This might become unpractical for the designer-toolmaker who without adequate scientific (and logistic) support can no longer balance demanding iterative processes within the planning of the overall product development process. It is therefore crucial that the designer or the design team is fully aware that the use of evolutionary principals and optimisation strategies should not exclude and limit the use of other creative design tools and methods, rather, those principals and strategies should enhance and streamline the creative behaviour, and be always directly related to the complexity of the object, as was demonstrated in this study where optimisation as a method was applied to the design of objects with the clear purpose of thermal control of interior spaces. This design goal can be obtained with the use of existing simulation software and with the computational power present in regular design offices. Furthermore, optimisation was used aside and intertwined with critical aesthetical evaluation. This is how a digital design process can be made interactive and how a creative and undertaking designer can tweak and hack to enforce vision and design intent. This is what is at the core of the processes described in this study, and these are the coordinates which oriented this research. All in all quite a different approach than similar ongoing research which concentrates more on the technical and mathematical evaluation of optimising: e.g. research such as developed by Brintrup et al. (2008), focused on optimising ergonomically a chair, but losing any designerly way of working.

The design space changes over (design) time, and new knowledge is constantly obtained and learned during the design process, at the same time constraints and requirements are constantly adaptation and reformulated, and it *should* be possible to adapt and reformulate them during the process. Therefore, adaption of the method to changing circumstances and reorganising the software construct by restructuring the combination of applied simulation and evaluation tools should also be possible and must be standard practice for the designer-toolmaker whenever optimisation becomes an integral part of the design process. The case studies showed that with the use and development of case-specific scripting by the designer, enough flexibility can be integrated in a software construct to allow for a highly iterative process directed to a desired and appropriate outcome.

Most of the research into the effective appliance of multi-objective optimisation in design is focussed on very specific and limited design problems. The Intelligent Genetic Design Tool (IGDT) is limited to the exploration of architectural trussed structural system (von Bulow, 2007). Although this IGDT is different from other approaches in computer aided structural design and claims to enhance the designer's creativity, it is based on a closed system with the consequent limited out-come, despite efforts to avoid design fixation. A similar limited outcome is also discussed in the work of Brintrup (Brintrup, Ramsden, Takagi, & Tiwari, 2008), and although the focus of this research is not on chair design but on the integration of qualitative and quantitative criteria in the design process (exemplified in this research by illustrating the design process of a chair), it clearly demonstrates the reductive effect of parametric constraints on truly creative results. She observes that "a truly ergonomic design process needs live experimentation; providing a completely computer –based ergonomic design optimisation framework is neither desirable nor possible" (Brintrup, Ramsden, Takagi, & Tiwari, 2008, p. 352). Her arguments and observations support the conclusions of this research project, when the goals of the proposed framework brings together qualitative and quantitative design criteria.

Digital design and this personalised development of digital tools to assist the design process will be an important characteristic of the future design process, as described and illustrated by various authors in Programming Cultures (Silver, 2006). As this research tried not to make a design *tool* but a design *method*, it is easily applied to different kinds of design projects, different strategies of designing or even different phases of the design process.

5.1 Limits on creativity

The optimisation process is computationally intensive and does take some considerable time if not executed on powerful systems. The actual time it takes for each optimisation cycle to produce evaluative results introduces as an immediate consequence a kind of intermittence in the design process. It can be argued that these interruptions restrain the creativity, especially in that phase of the design process where erratic and random exploration of solutions should or could be expected. This is however a rather narrow view of the use of optimisation in the design process, it has been stated in the beginning of this thesis that the objective is not researching the development of a design tool, but a design method, adaptable and applicable in conformity with the design process development strategy of the individual designer or architect. This implies that the development of the optimisation strategy, the selection of the analyses and simulation software programmes, the selection of the specific multi-objective optimisation algorithms and the proper construction and configuration of the software construct, are a creative endeavour by itself. Again, and in analogy with what has been the critique on the limits of existing CAD software programmes (see also 3.2.1) the proper optimisation construct, the methodology in itself, is, from the viewpoint of the designer as tool maker, heavily biased towards a perceived outcome. This is the reason why it should be possible, and why it is even desirable, that even starting with the same identical boundary conditions and specification, maybe even with a similar parametric construction, different designers will produce different solutions with comparable performative results. This, probably, can only be achieved if no limitations are imposed on the methodology itself, as was demonstrated in the case studies, and opposed to those optimisation tools which impose boundary conditions through the characteristics of the constructions of the proper software construct, and when the software construct cannot further be considered a construct of different softwares, but a fully integrated software itself, hence a design tool similar in construction approach to existing CAD tools .

This thesis describes a digital design method where the use of evolutionary principles is at the centre. The use of evolutionary algorithms however is not pacific among designers and architects. Some argue that applying 'evolution', as in its strict definition, is not at all a designerly way of working. They argue that the application of these techniques *replaces* design as it can be used to *breed* new forms rather than *design* them. Manuel de Landa in his essay "Deleuze and the use of the Genetic Algorithm in Architecture" (de Landa, 2001) challenges the use of evolutionary algorithms in the architectural design process and compares the role of the architect using those *virtual design tools* to the disappointing equivalent of a *breeder of virtual forms.* It is however not the objective of this research project to build a kind of architectural form breeding tool nor can the presented software construct be compared with a breeding strategy. And although it is true, just as de Landa recalls in his essay, that the new forms which emerge out of the evolutionary process are typically close to the original ones, the use of an optimisation strategy right in the beginning of the conceptual phase of the design process should assist the designer in exploring solutions by predicting possible aesthetical configurations, without enforcing the form of the final product. This research demonstrated that the possible outcome from the optimisation process depends heavily on the creative construction of the initial parametric geometry model. In the first case study, different constraints were progressively applied to limit the free form surface. The exploration of this free form geometry was limited to the manipulation of the coordinates of only 20 control points, and the surface was limited to a 5m x 5m base area. Under these conditions one could only expect small incremental improvement in the final outcome compared with the starting parametric geometry, and this is clear in the final set of optimised solutions. But all the intermediated solutions show that, even with such a basic parametric configuration, quite a few and different parametric reconfigurations are possible. The use of evolutionary systems in design can therefore only be justified if the designer can explore a solution space rich enough so that all the solutions cannot be predicted beforehand, which were the starting research objectives and which is illustrated in the case studies presented. Exploration based on parametrical geometries are at the same time (till now) the most indicated strategy for the correct use of optimisation algorithms based on evolutionary principles, but are also a serious limiting factor. This is not only demonstrated in the case studies in this research project, but is also described in other research (Turrin 2011). Thus future research should therefore be concentrated on integration of more complex parametric geometries and the incorporation of topological adaption of the objects.

5.2 Limits of optimisation

The main objective of this research study was building a framework for the integration of optimisation in the conceptual phase of the design process. Although focused on shading devices, this was merely an example and was used as just one possible design project of architectural objects. The main reason why shading devices were chosen for the case studies is because these devices can constitute an independent object of design in a building and can be optimised by at least two quantitative requirements.

The shading device used for the study of the optimisation construct in Case Study 2 is simple and functional. It was developed in order to comply with the goals of a study for multi-objective optimisation rather than for a genuine design project. Several concessions had to be made in order to match available computational power which is still very much a limitation on the practical use of optimisation methods in a design process. Other studies along similar strategies depend on the use of powerful servers (Turin, 2011)(Caldas, 2008) or networked, distributed computing was used (Deb, Zope & Jain, 2003).

More important than computing power, and crucial to the outcome of positive results is the manner how constraints are treated throughout the complete software construct, how those constraints are understood and interpreted, and how their influences on the final results and the esthetical outcome in particular, are recognised by the designer. According to Roozenburg and Eekels (1995) a designer deals with two kinds of constraints during the design process: those constraints which are knowingly applied to the design problem (input constraints) and those constraints which are unknowingly and due to habit enforced on the problem. The kinds of constraints imposed on the design problem and the quantity of constraints will limit and reduce the solution space available for exploration.

The present interest in research in optimisation and especially interactive optimisation combined with evolutionary algorithms (Brintrup et al., 2006) (Marin et al., 2008) is a result of the better understanding of the working and the pitfalls of the evolutionary algorithms and the perception of their role as a vital and essential part of a digital design process. And while optimisation is already widely used in the detailing phase of a design project with the very specific goal of complying with restricted technical requirements, the objective of contemporary research is focused on the introduction of optimisation strategies in the conceptual phase of the design process. It seems however very difficult not to focus on one specific object or system for optimisation and research results seem to confirm the allegation of de Landa that after the first generation evolutionary algorithms do not provide astonishing new results. A computational model for creative design based on collaborative interactive genetic algorithms (Banerjee, Quiroz, & Louis, 2008) focuses yet again on the optimisation of floor plans, an area pioneered by Radford and Gero (Radford & Gero, 1987) and although they use colours, widgets and a networked configuration of the optimisation routine, and the results of their study show that a collaborative optimisation of the layout of floor plans is ranked as more original than the results of individual optimisation, this is in fact a tool tailored specifically for resolving problems with the layout of floor plans not much different than a similar approach for the interactive optimisation of the layout of facilities in a shelter (Bénabès, Bennis, Poirson, & Ravaut, 2010) or the basic configuration of urban furniture (Machwe & Parmee, 2007).

Optimisation, *the act or process of making a design or a decision as fully perfect, functional or effective as possible* (Merriam-Webster, 2003, p. 871) as it is defined, does mean that the final objective of any such process must lead to only one solution. This might be (or must be) the desired outcome of any design problem, especially in performative design, but it is never certain that such a perfect proposal exists, nor that it eventually can be found. In a designerly way of thinking and solving problems one must acknowledge that simulation, and thus also analyses and evaluation, can only inform the designer with knowledge and explanations based on previous research of existing situations.

Optimisation or multi-objective optimisation is not a necessary component or strategy of a better design process nor is it exclusive for a digital design process. Which kind of design project can benefit from optimisation, in which stage of the design process can optimisation better be used, which particular algorithm can be adapted to a particular design problem and which objectives benefit from optimisation are all questions to be asked at the planning stage of a design process. Not all design projects benefit from interactive optimisation in the conceptual phase of the design process. However, making the right decisions and proceeding with the appropriate actions is the main reason why support given by scientific results is essential for designers. It is the aim of the designer for consistency and certainty that should support the use of optimisation in the design process: the potential to enhance the designers own creativity by discovering a good solution.

This research is based on case studies, and as such it can be argued that the proposed method can be applied in a general or universal way. But in design we are not dealing with a universal truth but with the extreme particular, the unique. This cannot be derived with inductive or deductive thinking alone. Design is a process of taking the right decisions and the appropriate actions, and in this process intuition will always keep on playing an important role much the same as in all other scientific endeavour.

It is necessary to streamline the design process: good and perfect solutions or proposals have to be presented within a reasonable time span. But deadline pressure and limited resources should not compromise the quality of the design process itself. If the perception of quality makes the project requirements more and more demanding and if environmental regulations turn the constraints into a quite complex set of rules, the design process does need specialised assistance.

Most of the knowledge used and generated by the computer construct described in this thesis, information such as direct solar gains results, stress analysis, thermal analyses, total area, etc., is objective and deductive knowledge. Gero (2007) argues that computational tools aim at encoding knowledge and making it available in an objective manner to the designer. And whilst most of the generated knowledge during a design process is in fact objective and deductive in nature, a special category of knowledge, which Gero calls first-person knowledge and which depends on the person and his interaction with the world, cannot be encoded or generated in the current computational paradigm. It is his opinion that computational tools are still restricted to these applications involving objective knowledge and cannot at present produce what is understood as designerly behaviour. The method and software construct described in this thesis aims to implement interpretation (by the human designer) and circumvent the limitations of the computational tool by forcing iterative behaviour into the design process.

This kind of methods and sophisticated digital design tools which allow personalisation and adaptation are desired by architects and designers and are a necessary complement for contemporary design problem solving where intuition, experience and first person knowledge are insufficient and not adequate. Different processes and different approaches are ongoingly researched (Peng & Gero, 2007) and will provide the framework for future development of similar methods and tools

5.3 Recommendations and Future Work

Recent research appoints to the difficulties for designers to address those design problems which involve more than two feedback loops (Love, 2009). An intuitive and emotionally based approach undeniably fails to address complex problems. Future research could be focussed on the integration of tools used to model and predict system behaviour into the optimisation software construct.

The software construct presented in this thesis does rely on the encoding of knowledge and making this knowledge available to the designer. This knowledge however is limited to so-called third-person knowledge (Gero, 2007). And although the system does produce first-person knowledge while interacting with the designer during the iterative loops, none of this knowledge is used for building constructive memory which can be used to produce computational systems with a designerly behaviour. Future research could focus on the development of such autonomous agents with enough flexibility and adaptability for inclusion of tailor-made design software constructs.

The use of multi objective evolutionary algorithms has definitely entered research in the field of architecture and design and is considered as a promising method to handle complexity issues related to all kinds of performative design. Some researchers are using fuzzy sets and fuzzy logic and models present at the output, information about the perceptual properties of a space (Bittermann, Sariyildiz, & Ciftcioglu, 2012).

In the case studies used to illustrate the FFO the generative approach to form finding was limited to the use of parametrical variables for geometry construction. And although the process is performance oriented the form finding process will always be restricted to the relationship between structure and geometry. Interesting complementary research would be the integration of material properties into the optimisation process, possibly along the framework of Variable Property Design (Oxman N., 2010), which would focus on behaviour, hence changing the form finding process into a behaviour finding process.

For many designers handmade sketches and drawings lack the rigour in their description and representation, necessary for exploring the technical and material specific possibilities of form and structure. However, the direct and exclusive use of digital tools within a digital environment is still experienced as unnatural and conflicting with the unstructured nature of the early phase of the design process. Further research in new methodologies into a seamless integration of physical into virtual processes, such as combining the speed of hand sketching and hand modelling to assess ideas with the simultaneous accuracy of virtual simulation and evaluation is promising area for further development.

In the time it took to complete this thesis digital design and especially digital form finding has become very popular among architects. In the world's leading universities research units are founded with very advanced research agendas, striking examples are "The Emergent Design Group" at the MIT and the "Hyperbody" research unit at the Delft University. During the last decade important graduate courses were also developed with the focus on the integration of advanced use of digital media in the architectural design process. The graduate course in "Emergent Technologies and Design" at the AA School of Architecture (2010) in London and the Master's Degree in Bio-digital Architecture at the University of Catalonia", are among those courses with almost ten years of experience. And if a Faculty of Architecture does not already provide a complete program in digital design, the educational curricula include lectures from visiting teachers or highly specialised workshops are organised around performative architecture, digital form finding, generative design computing, etc...

The leading CAD software packages for the architectural professions, such as Autodesk AutoCAD, Bentley Generative Components, ParaCloud and Rhino 3D are also developing add-on and plug-in tools targeted for a more creative and advanced use of the software. Rhino 3D provides an interesting tool with a smooth learning curve: Grasshopper allows for visual scripting and spaghetti wiring with astonishing results. Furthermore, highly specialised tools are being developed such as Galapagos, an evolutionary solver, Kangaroo, a live physics engine and GECO, a plug-in for quick thermal simulation with Ecotect, all for Rhino and Grasshopper. Attempts to couple generative design within a traditional CAD software program have also been investigated (Sivan, 2111) and will introduce this particular design technique to the mainstream users of CAD software.

As a final conclusion, one can state that the challenge to present computational constructs with a similar designerly behavior as a human designer is a major endeavor and a desired direction for future research: how can such a process start without all the necessary information, how will it continue after receiving new input; how can it be controlled and what part can be autonomous are all important questions for a research agenda. Furthermore, how can these constructs produce novel solutions or proposals, starting from the same (or very similar) requirements, at different moments and over again, much as can be expected from a specialist designer or architect. All this still needs profound reflection, and deeper ongoing research on the nature of the design process and in particular the digital design process.

The research work presented in this thesis, concentrates on the development of a method for the functional optimization of architectural objects where aesthetical considerations are a fundamental element driving an iterative process. Rapid prototyping and other techniques for building physical models and prototypes are essential for the evaluation and the validation of the selected and preferred solutions and are thus also a component of a digital design process. The nature of a digital design process allows for an easy transformation of the digital information describing an object to the exact references and information necessary to build a physical model using CNC machinery. The construction of full scale prototypes of some of the optimised solutions is therefore the objective of immediate further research work. Those physical models can then used for further testing and evaluation on the Test Cell existing on the campus of the university.

6 References

Abel, C. (2007). Virtual Evolution - A Memetic Critique of Genetic Algorithms in Design. *Presented at the 2007 Association of Architecture Schools Australasia Conference, Sydney*. Retrieved from http://hdl.handle.net/2100/461

Aicher, O. (1994). *Analogous Digital*. Berlin: Ernst & Sohn

Andersson, J. (2001). *A survey of multiobjective optimisation in engineering design*. (Technical Report Nº LiTH-IKP-R-1097 - Linkping University Sweden). Retrieved on March 15th, 2012 from website: neo.lcc.uma.es/emoo/andersson00.pdf.gz

Asadi, E., da Silva, M., Antunes, C., & Dias, L. (2012). Multi-objective optimisation for building retrofit strategies: A model and an application. *Energy and Buildings*, *44*, 81–87. doi:10.1016/j.enbuild.2011.10.016

Augenbroe, G. (2004). Trends in building simulation. In A. Malkawi, & G. Augenbroe (Eds.). *Advanced building simulation* (pp. 4-24). New York: Spon Press

Baeck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (1997). *Handbook of Evolutionary Computation*. Taylor & Francis.

Banerjee, A., Quiroz, J., & Louis, S. (2008). A model of creative design using collaborative interactive genetic algorithms. In J. Gero (Ed.). *Design Computing and Cognition '08* (pp. 397-416). Dordrecht: Springer Netherlands.

Bayazit, N. (2004). Investigating design: A review of forty years of design research. *Design Issues*, *20* (1), 16–29.

Bechthold, M. (2008). *Innovative surface structures: technologies and applications* (1st ed.). New York: Taylor & Francis.

Bénabès, J., Bennis, F., Poirson, E., & Ravaut, Y. (2010). An Interactive-based approach to the layout design optimisation. In A. Bernard (Ed.). *Global Product Development: Proceedings of the 20th CIRP Design Conference, Ecole Centrale de Nantes, Nantes, France, 19th-21st April 2010* (pp. 511-520). Springer. doi: 10.1007/978-3-642-15973-2_52

Benami, O., & Jin, Y. (2002). Creative stimulation in conceptual design. *ASME Conference Proceedings Volume 4: 14th International Conference on Design Theory and Methodology, Integrated Systems Design, and Engineering Design and Culture*.(pp. 251-263). doi: 10.1115/DETC2002/DTM-34023

Bilda, Z., & Gero, J. (2008). Idea development can occur using imagery only. In J. Gero (Ed.). *Design Computing and Cognition '08* (pp. 303-320). Dordrecht: Springer Netherlands.

Bittermann, S., Sariyildiz, I., & Ciftcioglu, O. (2012). A computational intellignece approach to alleviate complexity issues in design. In J. Portugali, H. Meyer, E. Stolk, & E. Tan (Eds.). *Complexity Theories of Cities have Come of Age* (pp. 347-369). Heidelberg: Springer Verlag.

Bollinger, K., Grohmann, M. & Tessmann, O. (2008). Form, Force, Performance Multi-Parametric Structural Design. *Versatility and Vicissitude - Architectural Design, 78* (2), 20-25. West Sussex: John Wiley & Sons Ltd.

Bordegoni, M., Colombo, G., & Formentini, L. (2006). Haptic technologies for the conceptual and validation phases of product design. *Computers & Graphics*, *30(3),* 377–390. doi:10.1016/j.cag.2006.02.012

Botha, M. & Sass, L. (2006). The Instant House: Design and digital fabrication of housing for developing environments, *CAADRIA 2006 Proceedings of the 11th International Conference on Computer Aided Architectural Design Research in Asia*. Retrieved from http://cumincad.scix.net/cgi-bin/works/Show?caadria2006_209

Brintrup, A., Ramsden, J., Takagi, H., & Tiwari, A. (2008). Ergonomic chair design by fusing qualitative and quantitative criteria using interactive Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, *12* (3), 343–354. doi:10.1109/TEVC.2007.904343

Brintrup, A., Takagi, H., Tiwari, A., & Ramsden, J. (2006). Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective optimization problems. *Journal of Biological Physics and Chemistry*, *6*, 137–146.

Buiding Smart. (2011). Model – Industry Foundation Classes (IFC). Retrieved from http://buildingsmart.com/standards/ifc/model-industry-foundation-classes-ifc

Burdek, B. (1991). *Design: Geschichte, Theorie und Praxis der* Produktgestaltung. Köln: DuMont.

Burnette, C. (2009). An Emotional Basis for Design Thinking. *Academia.edu*. Retrieved May 9, 2012, from http://independent.academia.edu/CharlesBurnette/Papers/192434/An_Emotional_Basis_for_Design_Thinking

Caldas, L. (2005). *An evolution-based generative design system: using adaptation to shape architectural form*. Doctoral disertation. Massachusetts Institute of Technology, Cambridge.

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

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

Carvalho, C., Lavareda, G., Lameiro, J., & Paulino, N. (2011). A step-up μ-power converter for solar energy harvesting applications, using Hill Climbing maximum power point tracking*. 2011 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1924–1927). doi:10.1109/ISCAS.2011.5937965

Chong, Y., Chen, C., & Leong, K. (2008). A heuristic-based approach to conceptual design. *Research in Engineering Design*, *20*(2), 97–116. doi:10.1007/s00163-008-0059-9

Choudhary, R., & Michalek, J. (2005). Design Optimization in Computer-Aided Architectural Design. *CAADRIA 2005 Proceedings of the 10th International Conference on Computer Aided Architectural Design Research in Asia* (pp. 149-159). Retrieved from http://cumincad.scix.net/cgi-bin/works/Show?_id=caadria2005_b_4a_b.

Christiaans, H. (1992). *Creativity in design: The role of domain knowledge in designing*. Utrecht: Lemma.

Ciftcioglu, Ö., & Bittermann, M. (2009). Adaptive formation of Pareto front in evolutionary multi-objective optimization. In Santos, W. (Ed.). *Evolutionary Computation* (pp. 417-444). Vienna: In-Tech.

Clancey, W. (1997). *Situated cognition*. Cambridge: University Press.

Crawley, D., Hand, J., Kummert, M., & Griffith, B. (2008). Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, *43(4),* 661–673. doi:10.1016/j.buildenv.2006.10.027

Cruz-Neira, C., Sandin, D., & DeFanti, T. (1993). Surround-screen projection-based virtual reality: the design and implementation of the CAVE. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. New York: Siggraph. 135-142.

Culha, B. (2005). An Outside-in Approach for Product Architecture and Design. In S. Andrew & L. William (Eds.). *ICED 05: International Conference on Engineering Design: Engineering Design and the Global Economy* (pp. 3352-3358). Barton, A.C.T.: Engineers Australia. Retrieved from http://search.informit.com.au/documentSummary;dn=400697342331477;res=IELENG

Damasio, A. (2006). *Descartes' Error: Emotion, Reason and the Human Brain*. Vintage.

Davarynejad, M. (2008). *Fuzzy Fitness Granulation in Evolutionary Algorithms for complex optimization*. Master Thesis. Ferdowsi University of Mashhad, Mashhad.

Daylight Factors: Ecotect. (n.d.). In *Ecotect Community Wiki*. Retrieved April 14, 2007, from http://wiki.naturalfrequency.com/wiki/Daylight_in_ECOTECT

Daylight Strategies. (n.d.). In *Ecotect Community Wiki*. Retrieved April 14, 2007, from http://wiki.naturalfrequency.com/wiki/Daylight_Strategies

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Deb, K., Zope, P., & Jain, A. (2003). Distributed Computing of Pareto-Optimal Solutions with Evolutionary Algorithms. In C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, & K. Deb (Eds.), *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science* (Vol. 2632, pp. 534-549). Springer Berlin/Heidelberg.

De Landa, M. (2001). *Deleuze and the Use of the Genetic Algorithm in Architecture*. Retrieved on January, 10, 2012 from http://crisisfronts.org/wp-content/uploads/2008/08/deluze_genetic-algorithm.pdf

Derelöv, M. (2009). *On Evaluation of Design Concepts: Modelling Approaches for Enhancing the Understanding of Design Solutions*. Linköping: Linköping University Electronic Press.

Dewey, J. (1896). The reflex arc concept in psychology. *Psychological Review, 3(4)*, 357–370.

Dominick, P., Demel, J., Lawbaugh, W., Freuler, R., Kinzel, G., & Fromm, E. (2000). *Tools and Tactics of Design.* John Wiley & Sons.

Donaldson, K. (2006). Product design in less industrialised economies: constraints and opportunities in Kenya. *Research in Engineering Design, 17* (3), 135–155. doi:10.1007/s00163-006-0017-3

Dorst, K. (2003). On the problem of design problems – problem solving and design expertise. *Journal of Design Research, 4* (2), 135–147. doi: 10.1504/JDR.2004.009841

Dorst, K. (2006). Design problems and design paradoxes. *Design issues*, *22* (3), 4–17.

Dorst, K. (2006). *Understanding Design (Revised).* Amsterdam: Book Industry Services (BIS).

Dorst, K., & Dijkhuis, J. (1995). Comparing paradigms for describing design activity. *Design Studies*, *16 (2),* 261–274. doi:10.1016/0142-694X(94)00012-3

Duarte, J., Rocha, J. ao M., & Soares, G. (2007). Unveiling the structure of the Marrakech Medina: A shape grammar and an interpreter for generating urban form. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *21*, 317–349. doi:10.1017/S0890060407000315

Dubberly, H. (2005). *How do you design?* Retrieved May 19, 2011, from http://www.dubberly.com/articles/how-do-you-design.html

Dunston, P., Arns, L., Mcglothlin, J., Lasker, G., & Kushner, A. (2011). An Immersive Virtual Reality Mock-Up for Design Review of Hospital Patient Rooms. In X. Wang, & J. Tsai (Eds.), *Collaborative Design in Virtual Environments* (pp. 167–176). Dordrecht: Springer Netherlands.

Eckert, C., Kelly, I., & Stacey, M. (1999). Cognitive Foundations for Interactive Generative Systems in Early Design. Presented at the International Conference on Engineering Design. Munich.

Eckert, C., Kelly, I., & Stacey, M. (1999). Interactive generative systems for conceptual design: An empirical perspective. *AI EDAM*, *13* (04), 303–320.

Fisher, T. (2008). *Designing (tools (for designing (tools for ...)))*. Doctoral Thesis. Faculty of the Constructed Environment, RMIT University Melbourne.

Fonseca, C. M., & Fleming, P. J. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, *3* (1), 1–16. doi:10.1162/evco.1995.3.1.1

Fontes, D. B. M. M., & Gaspar-Cunha, A. (2010). On Multi-Objective Evolutionary Algorithms. In C. Zopounidis & P. M. Pardalos (Eds.). *Handbook of Multicriteria Analysis* (pp. 287–310). Berlin Heidelberg: Springer Berlin / Heidelberg.

Foster, H. (2003). *Design and crime: and other diatribes*. New York: Verso.

Frazer, J. (2003). *De Identitate.* Retrieved July 11, 2008, from http://www.generativedesign.com/asialink/de6.htm

Gagne, J. M. ., & Andersen, M. (2010, August). *Multi-Objective Optimization for Daylighting Design Using a Genetic Algorithm*. Paper presented at SimBuild 2010 – 4th National Conference of IBPA-USA, New York. Retrieved from http://infoscience.epfl.ch/record/153674

Gallager, S., & M. (2009). Philosophical Antecedents of Situated Cognition. In P. Robbings & M. Ayudede (Eds.). *The Cambridge handbook of situated cognition* (pp. 35–52). Cambridge: University Press Cambridge. Retrieved from http://pegasus.cc.ucf.edu/~gallaghr/situated08.pdf

Gänshirt, C. (2007). *Tools for Ideas: Introduction to Architectural Design*. Birkhäuser GmbH.

Gaspar-Cunha, A. (2009). *Modelling and Optimisation of Single Screw Extrusion*. PhD Thesis. Köln, Germany: Lambert Academic Publishing.

Gaspar-Cunha, A., & Covas, J. (2004). RPSGAe — Reduced Pareto Set Genetic Algorithm: Application to Polymer Extrusion. In Gandibleux, X., Sevaux, M., Sörensen, K., & T'kindt, V. (Eds.). *Metaheuristics for Multiobjective Optimisation Vol.* 535 (pp. 221–249). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gedenryd, H. (1998). *How designers work - making sense of authentic cognitive activities*. Retrieved from http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=18828&fileOId=1484253.

Gero, J. (1998). Towards A Model Of Designing Which Includes Its Situatedness. In Grabowski, H., Rude, S., Grein, G. (Eds.). *Universal Design Theory*, Shaker. (pp. 47-55). Aachen: Verlag.

Gero, J. (2006). Understanding Situated Design Computing and Constructive Memory: Newton, Mach, Einstein and Quantum Mechanics. In Smith, I. (Ed.). *Intelligent Computing in Engineering and Architecture* (pp. 285–297). Springer.

Gero, J. (2007). Situated design computing: Principles. In Topping, B. (Ed.). *Civil Engineering Computations: Tools and Techniques* (pp. 25-35). Stirlingshire: Saxe-Coburg Publications.

Gero, J. S. (2007). Artificial intelligence in designing. *AI EDAM, 21* (1), 17–18.

Gero, J., & Kannengiesser, U. (2006). A framework for situated design optimization.. In van Leeuwen, J. & Timmermans, H. (Eds.). *Innovations in Design Decision Support Systems in Architecture and Urban Planning* (pp. 309–324). Berlin: Springer.

Gero, J., & Kannengiesser, U. (2009). An Ontology of Donald Shön's Reflection in Designing. *International Journal of Design Sciences and Technology, 15* (2), 77-90.

Gero, J., & Kelly, N. (2008). How can CAAD tools be more useful at the early stages of designing?: Towards Situated Agents That Interpret. In Nakapan, W., Mahaek, E., Teeraparbwong, K., & Nilkaew, P. (Eds.). *Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia* (pp. 309–316). Hong Kong: Chinese University of Hong Kong.

Glanville, R. (1994). Variety in design. *Systems Research*, *11* (3), 95–103. doi:10.1002/sres.3850110307

Goldschmidt, G., & Smolkov, M. (2006). Variances in the impact of visual stimuli on design problem solving performance. *Design Studies*, *27* (5), 549–569. doi:10.1016/j.destud.2006.01.002

Goldstein, J. (1999). Emergence as a Construct: History and Issues. *Emergence, 1* (1), 49–72. doi:10.1207/s15327000em0101_4

Gross, M. (1986). *Design as Exploring Constraints*. Doctoral dissertation. Massachussets Institute of Technology, Cambridge.

Gupta, S., & Okudan, G. E. (2008). Computer-aided generation of modularised conceptual designs with assembly and variety considerations. *Journal of Engineering Design*, *19* (6), 533–551. doi:10.1080/09544820802527106

Hemberg, M. (2011). Genr8 – A surface design tool [Computer Software]. Retrieved on March 14th, 2011, form http://projects.csail.mit.edu/emergentDesign/genr8/index.html

Hemberg, M., & O'Reilly, U. M. (2007). Integrating generative growth and evolutionary computation for form exploration. *Genetic Programming and Evolvable Machines, 8* (2), 163-186. doi: 10.1007/s10710-007-9025-y

Hensel, M. (2008). Performance-Orientated Design Precursors and Potentials. *Versatility and Vicissitude - Architectural Design, 78* (2), 48-53. West Sussex: John Wiley & Sons Ltd.

Hensel, M., & Menges, A. (2007). *Morpho-Ecologies: Towards Heterogeneous Space In Architecture Design*. London: AA Publications.

Hensel, M., & Menges, A. (2008). Versatility and Vicissitude: An Introduction to Performance in Morpho-Ecological Design. *Versatility and Vicissitude - Architectural Design, 78* (2), 6-12. West Sussex: John Wiley & Sons Ltd.

Hernandez, C. R. B. (2006). Thinking parametric design: introducing parametric Gaudi. *Design Studies*, *27* (3), 309–324. doi:10.1016/j.destud.2005.11.006

Hinterding, R. (1996). Gaussian mutation and self-adaption for numeric genetic algorithms. In *Proceedings of the IEEE International Conference on Evolutionary Computation, 1995* (1). doi:10.1109/ICEC.1995.489178

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Cambridge: MIT Press.

Horváth, I. (2005). On some Crucial Issues of Computer Support of Conceptual Design. In D. Talabă, D. & T. Roche (Eds.). *Product Engineering* (pp. 123–142). Dordrecht: Kluwer Academic. Publishers. Retrieved from http://www.springerlink.com/content/p273212643677656/

Huerta, S. (2006). Structural Design in the work of Gaudi. *Architectural Science Review*, *49* (4), 324–339.

Janssen, P. (2006). A generative evolutionary design method. *Digital Creativity*, *17*, 49–63. doi:10.1080/14626260600665736

Jansson, D. G., & Smith, S. M. (1991). Design fixation. *Design Studies*, *12* (1), 3–11. doi:10.1016/0142-694X(91)90003-F

Johnson, G., Gross, M. D., Hong, J., & Yi-Luen Do, E. (2007). Computational Support for Sketching in Design: A Review. *Foundations and Trends in Human-Computer Interaction, 2* (1), 1–93. doi:10.1561/1100000013

Jones, J. C. (1992). Design Methods (2nd ed.). John Wiley & Sons, Ltd.

Jonson, B. (2005). Design ideation: the conceptual sketch in the digital age. Design Studies, 26 (6), 613–624. doi:10.1016/j.destud.2005.03.001

Kalay, Y. E. (2004). Architecture's New Media: Principles, Theories, and Methods of Computer-*aided Design*. Cambridge: MIT Press.

Kalay, Y. E. (2006). The (changing) roles of computing in architectural design and education. Synthetic Landscapes. In *Proceedings of the 25th Annual Conference of the Association for Computer-Aided Design in Architecture,* 21-25. Retrieved from http://cumincad.scix.net/cgi-bin/works/Show?_id=acadia06_021&sort=DEFAULT&search=kalay&hits=73

Kalay, Y. E. (2006). The impact of information technology on design methods, products and practices*. Design Studies, 27* (3), 357–380. doi:10.1016/j.destud.2005.11.001

Kilian, A. (2006). *Design Exploration through Bidirectional Modeling of Constraints*. Doctoral dissertation. Massachusetts Institute of Technology, Cambridge.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220* (4598), 671–680. doi:10.1126/science.220.4598.671

Knight, T. (2004). *Interaction in Visual Design Computing*. Invited paper presented at the Visual and Spatial Reasoning in Design III, MIT.

Kolarevic, B. (2004). Back to the future: Performative Architectural Computing. *International Journal of Architectural Computing, 2* (01), pp. 44–50. doi:10.1260/1478077041220205

Kolarevic, B. (2005). *Architecture in the Digital Age: Design and Manufacturing* (New Ed.). Taylor & Francis Ltd.

Kolarevic, B., & Malkawi, A. (2004). *Performative Architecture: Beyond Instrumentality* (1st ed.). Routledge.

Koning, H., & Eisenberg, J. (1981). The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B*, *8.*, 295 – 323.

Kruger, C., & Cross, N. (2006). Solution driven versus problem driven design: strategies and outcomes. *Design Studies*, *27* (5), 527–548.

Kumar, A., Subramaniam, V., & Teck, T. (2000). Conceptual design of fixtures using machine learning techniques*. The International Journal of Advanced Manufacturing Technology, 16* (3), 176–181.

Kumar, S. (2006). Architecture and Industrial Design A Convergent Process for Design. . In *Proceedings of the 25th Annual Conference of the Association for Computer-Aided Design in Architecture,* 79-94. Retrieved from http://cumincad.scix.net/cgi-bin/works/Show?_id=acadia06_079&sort=DEFAULT&search=kumar&hits=10

Lawson, B. (2005). *How Designers Think: The Design Process Demystified* (4th ed.). Architectural Press.

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

Love, T. (2009). Counter-intuitive design thinking: implications for design education, research and practice. In L. Fennessy, R. Kerr, G. Melles, C. Thong, & E. Wright (Eds.). *Cumulus 38th South 2009, Nov 12, 2009*. Swinburne University of Technology & RMIT University.

Machwe, A. T., & Parmee, I. C. (2007). Supporting free-form design using a component based representation: an overview. *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation* (pp. 2821-2824). London: GECCO '07. doi:10.1145/1274000.1274096

Maciel, M. J. (2006). *Vitrúvio Tratado de Arquitectura*. Lisboa: IST Press.

Malkawi, A. (2004). Immersive building simulation. In A. Malkawi, & G. Augenbroe (Eds.). *Advanced building simulation* (pp. 217-246). New York: Spon Press.

Marin, P., Bignon, J. C., & Lequay, H. (2008). Generative exploration of architectural envelope responding to solar passive qualities. *Proceedings of the 26th eCAADe Conference on Education and Research in Computer Aided Architectural Design in Europe*. Antwerp: eCAADe.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization, 26* (6), 369–395.

Matthews, P. (2002). *The Application of Self Organizing Maps in Conceptual Design*. Doctoral dissertation. University of Cambridge, Fitzwilliam College.

Maver, T. (2000). A number is worth a thousand pictures. *Automation in Construction*, *9* (4), 333–336.

Mendonça, 2010. Low-span lightweight membranes in housing - environmental and structural potentialities. In P. Cruz (Ed.). *Structures and Architecture*. (pp. 349-350). Taylor & Francis Ltd.

Menges, A. (2006). Manufacturing Diversity. In A. Menges, M. Hensel, & M. Weinstock (Eds.) *Techniques and Technologies in Morphogenetic Design - Architectural Design*, 76 (2), 70-77. West Sussex: John Wiley & Sons Ltd.

Merriam-Webster's Collegiate Dictionary, 11th Edition (11th ed.). (2003). Springfield, Ma: Merriam-Webster.

Neil, M. (2006). The Sydney Opera House. *Structural Engineers Association of New York, 11* (1). Retrieved from http://www.seaony.org/publications/article.php?id=17.

Nishino, H., Takagi, H., Cho, S. B., & Utsumiya, K. (2001). A 3D modeling system for creative design. *Proceedings of the 15th International Conference on Information Networking* (pp. 479–486). Retrieved from IEEE CS Digital Library. doi.ieeecomputersociety.org/10.1109/ICOIN.2001.905468

Oosterhuis, Xia, & Hyperbody. (2009). Interactive Architecture #2. Episode Publishers. Retrieved from http://resolver.tudelft.nl/uuid:e3b17556-aa2a-43ae-a75c-39c8eb686566.

Oxman, N. (2010). *Material-based design computation*. Doctoral dissertation. Massachussets Institute of Technology, Cambridge. Retrieved from http://dspace.mit.edu/handle/1721.1/59192

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). Performance-based Design: Current Practices and Research Issues. *International Journal of Architectural Computing*, *6*, 1–17. doi:10.1260/147807708784640090

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.-H. (2006). *Engineering Design: A Systematic Approach* (3rd ed.). Springer.

Peng, W. & Gero, J. (2007a). Learning First-Person Knowledge through Interaction: Towards Effective Design Tools. In J. Bocquet (Ed.) *Proceedings of the16th International Conference on Engineering Design, ICED'07/561* (pp. 89-90). Paris: The Design Society.

Peng, W. & Gero, J.S. (2007b) Computer-aided design tools that adapt. In A. Dong, A. Vande Moere & J. S. Gero (Eds.), *CAADFutures2007* (pp. 417-430). Springer.

Radford, A. D., & Gero, J. S. (1987). Design by Optimization in Architecture, Building, and Construction. New York: Van Nostrand Reinhold Company.

Rao, S. S. (2009). Engineering Optimization: Theory and Practice (4th ed.). John Wiley & Sons,, Inc.

Reis, A. P. (2010). *Functional and Economical Evaluation of Lightweight Constructive Solutions*. Master Thesis. University of Minho, Guimarães.

Renner, G., & Ekárt, A. (2003). Genetic algorithms in computer aided design. *Computer-Aided Design*, *35* (8), 709–726. doi:10.1016/S0010-4485(03)00003-4

Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, *4* (2), 155–169.

Roberts, A., & Marsh, A. (2001). ECOTECT: Environmental Prediction in Architectural Education. *Architectural Information Management 19th eCAADe Conference Proceedings* (pp. 342-347). Retrieved: http://cumincad.scix.net/cgi-bin/works/Show?09cd.

Roozenburg, N. F. M., & Eekels, J. (1995). *Product Design: Fundamentals and Methods*. John Wiley & Sons.

Santos, A. (2001). *Desenvolvimento de uma metodologia de caracterização das conduções de iluminação natural nos edifícios baseada na avaliação "in situ".* Master Thesis. Faculty of Science at the Univesity of Lisbon, Lisbon.

Sarkar, S., Dong, A., & Gero, J. (2009). *A Problem Decomposition Method for Conceptual Design*. Presented at the International Conference on Research into Design, ICoRD'09, Bangelore. Retrieved from http://cs.gmu.edu/~jgero/publications/2009/09SarkarDongGeroICoRD.pdf

Sarkar, S., Gero,J., & Saunders, R. (2007). Re-Thinking Optimization as a Computational Design Tool: A Situated Agent Based Approach. Retrieved May 21, 2008, from http://cumincad.scix.net/cgi-bin/works/Show?_id=caadria2007_185&sort=DEFAULT&search=somwrita&hits=1

Sass, L. & Botha, M. (2006). The Instant House: A Model of Design Production with Digital Fabrication. *International Journal of Architectural Computing, 4* (4), 109-123. Retrieved from http://cumincad.scix.net/cgi-bin/works/Show?ijac20064407

Sass, L. (2007). A Palladian construction grammar—design reasoning with shape grammars and rapid prototyping. *Environment and Planning B: Planning and Design*, *34* (1), 87 – 106. doi:10.1068/b32071

Sass, L., & Oxman, R. (2006). Materializing design: the implications of rapid prototyping in digital design. *Design Studies*, *27* (3), 325–355. doi:10.1016/j.destud.2005.11.009

Schön, D. A. (1991). The Reflective Practitioner: How Professionals Think in Action (New ed.). Farnham: Ashgate Publishing Ltd.

Schön, D. A., & Wiggins, G. (1992). Kinds of seeing and their functions in designing. *Design Studies*, *13* (2), 135–156. doi:10.1016/0142-694X(92)90268-F

Schwede, D. A. (2006b). *Towards a Universal and Integrated Digital Representation of Physical Processes* . Doctoral dissertation. Faculty of Architecture, University of Sydney, Sydney. Retrieved from http://ses.library.usyd.edu.au/handle/2123/1433

Schwede, D.A. (2006a). A Digital Bridge for Performance-Based Design. In J. S. Gero (Ed.), *Design Computing and Cognition '06* (pp. 23–40). Dordrecht: Springer Netherlands.

Shea, K., & Cagan, J. (1999). The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent. *Design Studies*, *20* (1), 3–23. doi:10.1016/S0142-694X(98)00019-2.

Shea, K., & Smith, I. (2006). Improving Full-Scale Transmission Tower Design through Topology and Shape Optimization. *Journal of Structural Engineering, 132* (5), 781-790.

Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards integrated performance-driven generative design tools. *Automation in Construction*, *14* (2), 253–264. doi:10.1016/j.autcon.2004.07.002

Silver, M. (2006). Towards a Programming Culture in the Design Arts. In M. Silver (Ed.). *Programming cultures: art and architecture in the age of software - Architectural Design, 76* (1), 5-1. West Sussex: John Wiley & Sons Ltd.

Simon, H. (1996). *The sciences of the artificial*. MIT Press.

Smets, G. (1986). *Vormleer: De paradox van de vorm*. Amsterdam: Bert Bakker.

Sousa, P. & Duarte, J. (2005). Digital Desires, Material Realities. *Digital Design: The Quest for New Paradigms, 23th eCAADe Conference Proceedings* (pp. 221-228). Retrieved from http://cumincad.scix.net/cgi-bin/works/Show?2005_221.

Stiny, G., & Mitchell, W. J. (1978). The Palladian grammar. *Environment and Planning B: Planning and Design*, *5* (1), 5 – 18. doi:10.1068/b050005

Stiny, G., and J. Gips. 1972. Shape Grammars and the Generative Specification of Painting and Sculpture. In O. R. Petrocelli (Ed.). *The Best Computer Papers of 1971* (pp. 125-133). Philadelphia: Auerbach.

The Foundation of Wall and Ceiling Industry. (2009). *Building Information Modeling: Understanding and Operating in a New Paradigm*. Retrieved from http://www.awci.org/pdf/bim.pdf

Torres, S. L., & Sakamoto, Y. (2007). Facade design optimization for daylight with a simple genetic algorithm. *Proceedings of IBPSA International Building Performance Simulation Association* (pp. 1162-1167). Retrieved from http://www.ibpsa.org/m_papers.asp.

Tschimmel, K. (2010). Design as a Perception-in-Action Process. *Design Creativity 2010,* 223–230. Springer.

Tschimmel, K. C. (2010). *Sapiens e Demens no pensamento criativo do design*. Doctoral disertation. University of Aveiro, Aveiro.

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

U.S. Departement of Energy. (n.d.). *Passive Solar Home Design.* Retrieved April 30, 2012, from http://www.energysavers.gov/your_home/designing_remodeling/index.cfm/mytopic=10250

Ullman, D. G. (2009). *The Mechanical Design Process* (4th ed.). McGraw-Hill Higher Education.

Ulrich, K., & Eppinger, S. (2007). *Product Design and Development* (4th ed.). McGraw-Hill Higher Education.

Verdegay, J. L., Yager, R. R., & Bonissone, P. P. (2008). On heuristics as a fundamental constituent of soft computing. *Fuzzy Sets and Systems*, 159(7), 846–855. doi:10.1016/j.fss.2007.08.014

Vilfredo Pareto. (2012). In *Encyclopædia Britannica Online*. Retrieved from http://www.britannica.com

von Bulow, P. (2007). *An intelligent genetic design tool (IGDT) applied to the exploration of architectural trussed structural systems*. Doctoral dissertation. Universitat Stuttgart, Stuttgart.

Wang, J. (2002). Improved engineering design concept selection using fuzzy sets. *International Journal of Computer Integrated Manufacturing*, *15* (1), 18–27. doi:10.1080/09511920110034996

Wynn, D. C., Eckert, C. M., & Clarkson, P. J. (2007). Modelling iteration in engineering design. In *Proceedings of the16th International Conference on Engineering Design, ICED'07/561* (pp. 1-12). Paris: The Design Society.

Ye, J., & Campbell, R. I. (2006). Supporting conceptual design with multiple VR - based interfaces. *Virtual and Physical Prototyping, 1* (3), 171-181.

Yen, J., Liao, J. C., Bogju Lee, & Randolph, D. (1998). A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *28* (2), 173–191. doi:10.1109/3477.662758

Zadeh (n.d.). BISK - The Berckeley Initiative in Soft Computing. Retrieved on March 14th, 2011, from http://www.cs.berkeley.edu/~zadeh/

Zeiler, W, Savanovic, P. & Quanjel, E.. 2006. Design Decision Support for the Conceptual Phase of the Design Process. IASDR 07 – International Association of Societies of Design Research. Hong Kong. Retrieved from http://www.sd.polyu.edu.hk/iasdr/proceeding/papers/Design%20Decision_%20support%20for%20the%20conceptual%20phase.pdf

6.1 *References to Software and Websites*

Avida (website: http://avida.devosoft.org/about/ – last accessed on April 4, 2011)

Bentley's Generative Components (website: http://www.bentley.com/en-US/ - last accessed on May 5, 2011)

Catia (website: http://www.3ds.com/products/catia - last accessed on April 4, 2011)

Emergent Design Group - EDG (website: http://mit.edu/edgsrc/www/ - last accessed on November 10, 2011)

Emergent Technologies and Design Course – AA London (website: http://www.aaschool.ac.uk/STUDY/GRADUATE/emtech.php - last accessed on Dezember 10, 2011)

Energy Plus 7.0 (website: http://apps1.eere.energy.gov/buildings/energyplus/ - last accessed on October 10, 2011)

Grashopper Rhinoceros 3D plug-in (website: http://www.grasshopper3d.com/ - last accessd on November 10, 2011)

Hyperbody (website: http://www.hyperbody.nl/ - last accessed on November 10, 2011)

Master's Degree in Biodigital Architecture (website: http://www.uic.es/progs/obj.uic?id=4d904c010dce1 – last accessed on Dezember 11, 2011)

Maya Embedded Language for the Autodesk Maya Software (website: http://usa.autodesk.com/maya/ - last accessed November 10, 2011)

Maya Scripting (website: http://mayamel.tiddlyspot.com/ - last accessed on May 4, 2011)

Microsoft Visual Basic 2008 (website: http://msdn.microsoft.com/en-us/express/future/bb421473 - last accessed on January 18, 2012)

Microstation (website: http://www.bentley.com/en-US/Products/MicroStation/ - last accessed on July 4, 2011)

ParaCloud Modeler (website: http://www.paracloud.com/ - last accessed on July 5, 2011)

Processing (website: http://processing.org/ - last accessed on November 5, 2011)

Rhinoceros 3D (website: http://www.rhino3d.com/ - last accessed on March 11, 2011)

SoldidWorks (website: http://www.solidworks.com/ - last accessed on July 7, 2011)

ANNEX A

The Software Construct Setup

The main objective of this research experiment is to develop and test a method which introduces the conecpt of optimisation in the design process. This implies the building of a software construct which assists the designer in the conceptual phase of a design process. This software construct is build purposefull and especific according to design intent, design objectives and personal characteristics and team expertise.

The software construct built for this study consists of core script in Visual Basic which controls the building of a geometry in CAD software Rhino 3D using an other script written in Rhino Script (a scripting language based on VB developed by the makers of Rhino 3D) and information about the coordinates of the control points of a NURBS surface stored in a file. The outcome of this first part of the script is a geometry in the exact file format for import in the simulation software Ecotect 2011. The core script controls the execution of a script written in Lua scripting language which performs the different analysis tasks within the simulation software. The results are written to a file for use by the optimisation algoritm. Optimisation of the results was realised using an optimisation algorithm written in C++, originally developed by Gaspar Cunha (1999) and adapted by him especially for this research project. The optimisation algorithm creates the Pareto frontier graphs which are used in this study for analyzing the results, but also produces the necessary information the core script to construct and generate a visual presentation of all the non-dominated solutions of a particular optimisation run. A GUI was written in Visual Basic 2008 Express for a better and more interactive personal and subjective evaluation of the designer.

Vbscript Main Structure

```
FILE NAME = v3_0 Generic Vulcano OptimisationModel.vbs
Dirk Loyens/Ferrie van Hattum 11012011
'-------------------------------------------------------------------------------
NOT PREPARED FOR RUNNING ON 64bit-COMPUTER
This script will simulate the MOEA and create identical cones for a shading device grid
Script will run as much iteration as specified in the StartCycleCS2.txt file
The script will only function with 25 modules.

If used in optimisation construct disable the previous mentioned parts.
The front circle is randomly controlled by 8 control points
each control point moves completely randomly within a confined space with an upper
and lower limit for all coordinates

Be carefull to have your PenDisk set as a Z:\ drive.
No Thermal Gains Calculated Yet - Only average lighting in Lux
No prevention of surface intersection and module intersection as yet
Directories necessary:
  - ArchiveCS2 (for storing the results)

STILL TO CHECK Files necessary in the root of the Zpen:
  - Check.txt
  - StartCycleCS2.txt (in the root of the Zpen with the number of cycles of the main
loop – whenever run in simulation)
  - Test.txt
  - Xcoord.txt
  - EcoPanelScript.scr (the original script in Lua which runs in Ecotect)
  - BaseCube.eco (Ecotect file with the cube prepared for calculations)
Important: change line xxx and xxx to the exact location of your Ecotect Application
-------------------------------------------------------------------------------

'necessary constant defined once in the beginning of the script
  Const ForAppending = 8
  Const ForReading = 1
  Const ForWriting = 2

'minimise all windows
   dim objShell
   set objShell = CreateObject("Shell.Application")
   objShell.MinimizeAll
   set objShell = nothing


                           ONLY FOR INDEPENDENT RUN
         defining the variables
         Dim Row
         Dim Modules
         Dim v

'create a XLS file to store coordinates and to store calculated values
  Set objXL = CreateObject("Excel.Application")
  objXL.Visible = False
  objXL.Workbooks.add
```

```
Dim k: k=1
For k = 1 to 55
    objXL.Columns(k).ColumnWidth = 10
 next

        objXL.Columns(56).ColumnWidth = 25
        objXL.Cells(1, 56).Value = "FileName"
        k=0
        l=1
        For k = 1 to 27 step 3
        objXL.Cells(1, k+1).Value = "PC X" & l
        objXL.Cells(1, k+2).Value = "PC Y" & l
        objXL.Cells(1, k+3).Value = "PC Z" & l
        l=l+1
        Next
        l=1
        For k = 28 to 53 step 3
        objXL.Cells(1, k+1).Value = "PR X" & l
        objXL.Cells(1, k+2).Value = "PR Y" & l
        objXL.Cells(1, k+3).Value = "PR Z" & l
        l=l+1
        Next

      objXL.Range("A1:BD1").Select
      objXL.Selection.Font.Bold = True
      objXL.Selection.Interior.ColorIndex = 1
      objXL.Selection.Interior.Pattern = 1
      objXL.Selection.Font.ColorIndex = 2
      objXL.Range("A1:BD200").Select
      objXL.Selection.HorizontalAlignment = -4108

        'creates unique file name for the XLS file
        Dim strXLSRef
        strXLSRef = CStr(Year(Date)) & CStr(Month(Date)) & CStr(Day(Date)) & "_" _
        & Cstr(Hour(Now)) & Cstr(Minute(Now))

        'Saves a Excel file
        objXL.ActiveWorkbook.SaveAs("Z:\" & strXLSRef & ".xlsx")
        objXL.Workbooks.Close
        objXL.Quit
```

DEFINING THE TOTAL DESIRED SIMULATED ITERATIONS – THIS PART IS NOT NECESSARY FOR FINAL OPTIMISATION RUN

```
        'read from the StartCycleCS2.txt file the total of desired simulated iterations
        the script will run
        Dim Iterations
        Set objFSO = CreateObject("Scripting.FileSystemObject")
        Set objFile = objFSO.GetFile("Z:\StartCycleCS2.txt")
        If objFile.Size > 0 Then
                Set objReadFile = objFSO.OpenTextFile("Z:\StartCycleCS2.txt",1)
                Iterations = objReadFile.ReadAll
                objReadFile.Close
                Else
```

```
        Wscript.Echo "No information about iteration cycles in file
        StartCycleCS2.txt"
End If

Set objFSO = nothing
Set objFile = nothing

'Starting the main loop in the script
Dim ScriptCycle
For ScriptCycle = 1 to Iterations
'ScriptCycle = 1   'only if not running multiple iterations
```

CREATE A TXT FILE WITH RANDOM VALUES – THIS PART IS NOT NECESSARY FOR FINAL OPTIMISATION RUN

```
Dim MaxYZ: MaxYZ = 500  'final rnd number will vary between -250 and 250
Dim MinYZ: MinYZ = 0
Dim MaxX
MaxX = 20          'too much variation does not result in feasible solutions
Dim MinX: MinX = 0
'Dim MaxRow: MaxRow = 4          'max. 36 modules
'Dim MinRow: MinRow = 2          'min 4 modules

Dim strData
ReDim arrRndNum(24)     'array with the random row count and all the
                        random coord of the 8 control points

Randomize
'arrRndNum(0) = Int((MaxRow - MinRow + 1)*Rnd()) + MinRow
arrRndNum(0) = 5 'total number of modules is fixed on 25

For i = 1 To 24 step 3
arrRndNum(i) = Int((MaxX - MinX + 1)*Rnd()) + MinX          'rnd X coord
arrRndNum(i+1) = Int((MaxYZ - MinYZ + 1)*Rnd()) + MinYZ     'rnd Y coord
arrRndNum(i+2) = Int((MaxYZ - MinYZ + 1)*Rnd()) + MinYZ     'rnd Z coord
Next

'write the random values to CircleCoordCS2.txt file
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.OpenTextFile("Z:\CircleCoordCS2.txt",2,true)
For j = 0 to 24
        strData = arrRndNum(j)
        objFile.WriteLine(strData)
Next

objFile.Close
```

END OF THE "ONLY FOR INDEPENDENT RUN" PART

PART ONE – RUNNING THE MAIN LOOP OF THE SCRIPT – ALSO FOR FINAL OPTIMISATION RUN

```
'creates unique file name for reference by counting seconds since midnight combined with the date of creation of the file

	Dim strURef
	strURef = CStr(Year(Date)) & CStr(Month(Date)) & CStr(Day(Date)) & "_" & Cstr(Hour(Now)) & Cstr(Minute(Now)) & "_" & Cstr(Second(Now))

	'reading all the data from the txt file in an array arrLine
	Dim objFSO, objTextFile
	Set objFSO = CreateObject("Scripting.FileSystemObject")
	Set objTextFile = objFSO.OpenTextFile("Z:\CircleCoordCS2.txt", ForReading)
	ReDim arrLine(24)
	  For k = 0 to 24
		arrLine(k) = objTextFile.Readline
	  Next
	objTextFile.Close
	Set objFSO = Nothing

	'preparing the data for use in Rhino

	Dim Row: Row = arrLine(0)	-- get the random number of rows of modules
	Dim Modules: Modules = Row * Row		-- defines how many modules
	Dim v: v=Int(2400/Sqr(Modules))		-- value to calculate number of modules

	Dim Xcoord: Xcoord = Int(v-(v/3))	-- correction for the depth of the vulcano
	Dim CorLim: CorLim = MaxYZ/2		-- correction to set rnd numbers between
						positive and negative limits

ReDim arrPointC(8) 'set the coordinates of the circle points
arrPointC(0)=Array(Cint(arrLine(1))+Xcoord,(v/6)+Cint(arrLine(2))-CorLim,(v/6)+Cint(arrLine(3))-CorLim)
arrPointC(1)=Array(Cint(arrLine(4))+Xcoord,(v/2)+Cint(arrLine(5))-CorLim,(v/8)+Cint(arrLine(6))-CorLim)
arrPointC(2)=Array(Cint(arrLine(7))+Xcoord,(5*v/6)+Cint(arrLine(8))-CorLim,(v/6)+Cint(arrLine(9))-CorLim)
arrPointC(3)=Array(Cint(arrLine(10))+Xcoord,(7*v/8)+Cint(arrLine(11))-CorLim,(v/2)+Cint(arrLine(12))-CorLim)
arrPointC(4)=Array(Cint(arrLine(13))+Xcoord,(5*v/6)+Cint(arrLine(14))-CorLim,(5*v/6)+Cint(arrLine(15))-CorLim)
arrPointC(5)=Array(Cint(arrLine(16))+Xcoord,(v/2)+Cint(arrLine(17))-CorLim,(7*v/8)+Cint(arrLine(18))-CorLim)
arrPointC(6)=Array(Cint(arrLine(19))+Xcoord,(v/6)+Cint(arrLine(20))-CorLim,(5*v/6)+Cint(arrLine(21))-CorLim)
arrPointC(7)=Array(Cint(arrLine(22))+Xcoord,(v/8)+Cint(arrLine(23))-CorLim,(v/2)+Cint(arrLine(24))-CorLim)
arrPointC(8)=Array(Cint(arrLine(1))+Xcoord,(v/6)+Cint(arrLine(2))-CorLim,(v/6)+Cint(arrLine(3))-CorLim)

	ReDim arrPointR(8) 'set the coordinates of the square points
	arrPointR(0)=Array(0,0,0)
	arrPointR(1)=Array(0,(v/2),0)
	arrPointR(2)=Array(0,v,0)
	arrPointR(3)=Array(0,v,(v/2))
	arrPointR(4)=Array(0,v,v)
	arrPointR(5)=Array(0,(v/2),v)
	arrPointR(6)=Array(0,0,v)
```

```
arrPointR(7)=Array(0,0,(v/2))
arrPointR(8)=Array(0,0,0)
```

ONLY FOR INDEPENDENT – NOT FOR FINAL OPTIMISATION RUN

```
'write the random values to an XLS file
XLrow = ScriptCycle+1
Set objXL = CreateObject("Excel.Application")
Set objWorkbook = objXL.Workbooks.Open("z:\" & strXLSRef & ".xlsx")
objXL.Visible = False

objXL.Cells(XLrow, 1).Value = arrRndNum(0)
objXL.Cells(XLrow, 56).Value = strURef
p=0
For j = 1 to 24 step 3
        objXL.Cells(XLrow, j+1).Value = arrPointC(p)(0)
        objXL.Cells(XLrow, j+2).Value = arrPointC(p)(1)
        objXL.Cells(XLrow, j+3).Value = arrPointC(p)(2)
        If p=0 Then
        objXL.Cells(XLrow, j+25).Value = arrPointC(p)(0)
        objXL.Cells(XLrow, j+26).Value = arrPointC(p)(1)
        objXL.Cells(XLrow, j+27).Value = arrPointC(p)(2)
        End If
        p=p+1
Next
p=0

For j = 28 to 51 step 3
        objXL.Cells(XLrow, j+1).Value = arrPointR(p)(0)
        objXL.Cells(XLrow, j+2).Value = arrPointR(p)(1)
        objXL.Cells(XLrow, j+3).Value = arrPointR(p)(2)
        If p=0 Then
        objXL.Cells(XLrow, j+25).Value = arrPointR(p)(0)
        objXL.Cells(XLrow, j+26).Value = arrPointR(p)(1)
        objXL.Cells(XLrow, j+27).Value = arrPointR(p)(2)
        End If
        p=p+1
Next

'save and close Excel
objXL.ActiveWorkbook.Save
Set WshShell = CreateObject("WScript.Shell")
        wscript.sleep 500
objXL.Quit

'run Rhino to create a new object for simulation with EcoTect
Set Rhino = CreateObject("Rhino4.Application")

'wait until Rhino has started
strProcName = "Rhino4.exe"
strComputer ="."
Set WshShell = CreateObject("WScript.Shell")
Set objWMIService =_
GetObject("winmgmts:{impersonationLevel=impersonate," &_
"authenticationLevel=pktPrivacy}!//" & strComputer & "\root\cimv2")
```

```
Do until IsStarted = True
            WScript.Sleep 1000
            Set colProcesses = objWMIService.ExecQuery("Select * from_
            Win32_Process Where Name = '" & strProcName & "'")
            For Each objProcess in colProcesses
            IsStarted = True
            exit For
            Next
Loop

'pause 1 second extra
wscript.sleep 1000

'enable VB to interact with Rhino
Set RhinoScript = Rhino.GetScriptObject()

'set working folder to pen
Dim penZ
penZ = "z:\"
Rhinoscript.WorkingFolder penZ

'ACTIVATE - show Rhino on the screen
            Rhino.Visible = False
            'Rhino.Visible = True
            'maximises the perspective vieuwport
            RhinoScript.Command "_-MaxViewport "
            'zoom to window
            RhinoScript.ZoomExtents

'build the circle
RhinoScript.AddCurve(arrPointC)
Dim strObjectC
strObjectC = RhinoScript.FirstObject

'build the square
RhinoScript.AddPolyline(arrPointR)
Dim strObjectR
strObjectR = RhinoScript.FirstObject

'loft the vulcano
ReDim arrObjects(1)
arrObjects(0)=strObjectC
arrObjects(1)=strObjectR
RhinoScript.AddLoftSrf(arrObjects)
Dim strObjectA
strObjectA = RhinoScript.FirstObject

'copy the element to fill a square of 2400 x 2400
Dim numPanel,colPanel
Dim y: y=0
Dim z: z=v          'skip the first module already build
ReDim arrEnd(Modules)
Dim arrObjs
arrObjs = Array (strObjectA)
Dim arrStart
arrStart = Array(0,0,0)
```

```
For numPanel = 2 To Modules
arrEnd(numPanel)=Array(0,0+y,0+z)
RhinoScript.CopyObjects arrObjs, arrStart, arrEnd(numPanel)
colPanel=(numPanel)/(Sqr(Modules))-Int((numPanel)/(Sqr(Modules)))
If colPanel>0 Then z=z+v
If colPanel=0 Then z=z-(2400-v):y=y+v
Next

'copy the txt file with new name to archiveCS2 directory
Dim strNFile, strTxtFile
strTxtFile = "Z:\CircleCoordCS2.txt"
strNFile = "Z:\ArchiveCS2\" & strURef & "_CircleCoordCS2.txt"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFileCopy = objFSO.GetFile(strTxtFile)
objFileCopy.Copy (strNFile)
Set objFSO = nothing

'overwrites the base files which serve for analysis in the proper software
with data from this loop
Rhinoscript.WorkingFolder penZ
RhinoScript.Command "SelAll"
RhinoScript.Command "_-Export Z:\EcoVulcanoCS2.3ds enter"
wscript.sleep 500

'copy the 3ds file with new name to archiveCS2 directory
Dim strNFile2, strTxtFile2
strTxtFile2 = "Z:\EcoVulcanoCS2.3ds"
strNFile2 = "Z:\ArchiveCS2\" & strURef & "_EcoVulcanoCS2.3ds"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFileCopy = objFSO.GetFile(strTxtFile2)
objFileCopy.Copy (strNFile2)
Set objFSO = nothing

'save Rhinofile with unique name to archiveCS2 directory
Dim strRhino
strRhino = "Z:\ArchiveCS2\" & strURef & "_Rhino.3dm"
RhinoScript.Command "_-Save " & strRhino

'copy pix file with the new name to archiveCS2 directory
Dim archJpg, strView
archJpg = "z:\ArchiveCS2\" & strURef & ".jpg enter"
Rhino.Visible = True
strView = RhinoScript.CurrentView
RhinoScript.ZoomExtents strView
RhinoScript.ShowGridAxes, False
RhinoScript.ShowGrid, False
wscript.sleep 500
RhinoScript.ViewDisplayMode strView, 1
RhinoScript.Command "_-ViewCaptureToFile " & archJpg
wscript.sleep 500
Rhino.Visible = False

'finish RhinoScript
RhinoScript.Exit ()

'wait an extra 4 seconds
```

```
WScript.Sleep 4000

'write the number 0 to a file - if the ecoscript hangs this zero will not be
overwritten and the results will be removed
Dim strCheckFile, strCheck
strCheck = 0
strCheckFile = "Z:\Check.txt"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objWriteFileA = objFSO.CreateTextFile(strCheckFile)
objWriteFileA.WriteLine strCheck
objWriteFileA.Close
Set objFSO = nothing

'message for verification - can be deleted from script
Set WshShell = CreateObject("WScript.Shell")
WshShell.Popup "Verification - Starting Ecotect " & Cycle & "!", 2, , 64
set WshShell = nothing

                    PART TWO - ECOTECT SIMULATION

Dim WshShell, oExec, bExex
Set WshShell = CreateObject("WScript.Shell")

'open Ecotect and open EcoPanelScript in ScriptManager
Set oExec = WshShell.Exec("C:\Program Files\Autodesk\Ecotect Analysis
2011\Ecotect.exe")
WScript.Sleep 10000
Set bExec = WshShell.Exec("C:\Program Files\Autodesk\Ecotect Analysis
2011\ScriptManager.exe EcoVulcanoScript.scr")

'wait until Scriptmanager has started
WScript.Sleep 10000       'on faster computers waiting can be reduced
WshShell.Run "Z:\ActivateECOscript.exe"
WScript.Sleep 2000        'on faster computers waiting can be reduced
WshShell.SendKeys "^{F9}"        'start EcoPanelScript.scr


'wait until ecoscript has finished
WScript.Sleep 20000

Dim Check
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.GetFile("Z:\Check.txt")
Set objReadFile = objFSO.OpenTextFile("Z:\Check.txt",1)
        Check = objReadFile.ReadAll
        objReadFile.Close

'if nothing has been written to the file than Ecotect failed and nothing has
been calculated
'Ecotect will be closed and nothing will be calculated because the results are
not valuable

If Check = 0 then
        Dim objWMIService, objProcess, colProcess
        Dim strComputer, strProcessKill, strProcessKill2
        strComputer = "."
```

```
        strProcessKill = "'ScriptManager.exe'"
        strProcessKill2 = "'Ecotect.exe'"

        Set objWMIService = GetObject("winmgmts:" & {impersonation_
        Level=impersonate}!\\" & strComputer & "\root\cimv2")
        Set colProcess = objWMIService.ExecQuery ("Select * from_
        Win32_Process Where Name = " & strProcessKill )
        For Each objProcess in colProcess
        objProcess.Terminate()
        Next
        Set colProcess = nothing

        Set colProcess = objWMIService.ExecQuery ("Select * from_
        Win32_Process Where Name = " & strProcessKill2 )
        For Each objProcess in colProcess
        objProcess.Terminate()
        Next
        Set colProcess = nothing
End If

'if something has been written to the file, Ecotect is functioning and will
produce valuable results
'and the script will be waiting till the end of the calculations by checking the
existance of a file

Set objFSO = CreateObject("Scripting.FileSystemObject")
If Check = 1 then
While objFSO.FileExists ("z:\Test.txt") = False
WshShell.Popup "Ecotect calc running: please, wait a moment!", 2, , 64
WScript.Sleep 5000
Wend
dim demofile
set demofile = objFSO.GetFile("z:\Test.txt")
demofile.Delete
End If

Set objFSO = nothing
Set objFile = nothing

'write the Average result value to the archiveCS2 directory
Dim strAvFile, strPavFile
strAvFile = "Z:\Average.txt"            'ecoscript does not function with a
different path in the filename average has to stay on the root of the pen
strAavFile = "Z:\ArchiveCS2\" & strURef & "_centerLux.txt"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFileCopy = objFSO.GetFile(strAvFile)
objFileCopy.Copy (strAavFile)
Set objFSO = nothing

'end of Ecotect simulation
strComputer = "."
Set objWMIService = GetObject("winmgmts:" &_
"{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")

Set colProcessList = objWMIService.ExecQuery("Select * from Win32_Process_
Where Name = 'ScriptManager.exe'")
```

```
For Each objProcess in colProcessList
            objProcess.Terminate()
Next

Set colProcessList = objWMIService.ExecQuery("Select * from Win32_Process_
Where Name = 'Ecotect.exe'")
For Each objProcess in colProcessList
            objProcess.Terminate()
Next

'end of the main loop of the script
next
```

PART THREE – FINISHING ALL THE PROCESSES

```
Dim strProcessKill11
strComputer = "."

strProcessKill11 = "'Rhino4.exe'"

Set objWMIService = GetObject("winmgmts:" &
"{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set colProcess = objWMIService.ExecQuery ("Select * from Win32_Process
Where Name = " & strProcessKill11 )
For Each objProcess in colProcess
objProcess.Terminate()
Next
Set colProcess = nothing


'End of Script Message

Set WshShell = CreateObject("WScript.Shell")
WshShell.Popup "End of Script!", 4, , 64
WScript.Quit
```

END OF SCRIPT

Ecotect Script

```
Dirk Loyens 18012011 – direct solar gains for two Julian days
and average daylight in lux over 3 nodes
-------------------------------------------------------------------------------
cmd("app.activate")
cmd("calc.adjacencies", 2500, true)

filename1 = "z:\\direct solar gains summer.txt"        -- open file to write results to
file1 = openfile(filename1, "w")
total1 = 0

filename2 = "z:\\direct solar gains winter.txt"        -- open file to write results to
file2 = openfile(filename2, "w")
total2 = 0

filename3 = "z:\\average daylight in lux.txt"
file3 = openfile(filename3, "w")

set("dayoftheyear", 172)                           -- set Julian day according to table
cmd("calc.thermal.gains", 1)                       -- calc thermal gains for zone 1
for hr = 0,23 do
dsg1 = get("results.array", 3, hr)                 -- get direct solar gains from table = 3
output = dsg1
write(file1,output, "\n")                          -- write total result to file
print("direct solar gains on hour ", hr, dsg1)    -- print results on screen
total1 = total1 + dsg1                             -- calculate the total result for that day
end
output = total1
write(file1,output, "\n")                          -- write total result to file
print("\n", "total", total1)                       -- print the total result for that day
closefile(file1)

set("dayoftheyear", 355)                           -- set Julian day according to table
cmd("calc.thermal.gains", 1)                       -- calc thermal gains for zone 1

for hr = 0,23 do
dsg2 = get("results.array", 3, hr)                 -- get direct solar gains from table = 3
output = dsg2
write(file2,output, "\n")                          -- write total result to file
print("direct solar gains on hour ", hr, dsg2)    -- print results on screen
total2 = total2 + dsg2                             -- calculate the total result for that day
end
output = total2
write(file2,output, "\n")                          -- write total result to file
print("\n", "total", total2)                       -- print the total result for that day
closefile(file2)

set("calc.windows", 0)                             -- cleanliness = 1
set("calc.sky", 0, 6500)                           -- luminance for Guimarães
set("calc.precision", 2)                           -- precision set to high
cmd("calc.lighting.grid daylight", false, 0)
set("grid.data", 0)
```

```
dl1 = get("grid.cell", 0, 0, 0)
dl2 = get("grid.cell", 1, 1, 0)
dl3 = get("grid.cell", 0, 2, 0)
dl4 = dl1 + dl2 + dl3
dl = dl4 / 3
output = dl
write(file3,output, "\n") -- shrijf resultaat naar file
print("daylight = ", dl) -- print resultaten ook op het scherm
closefile(file3)
```

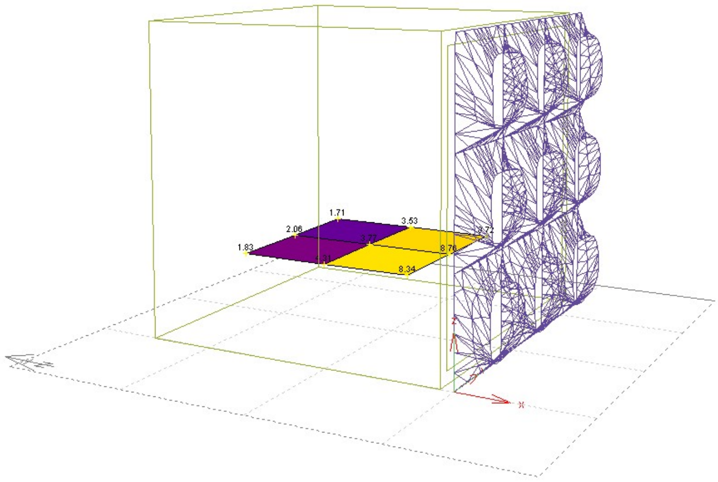
ANNEX B

EcoTect 2011 calculates the daylight factor according to the Building Establishment Split-Flux method. This method assumes that the quantity of natural light which reaches any point inside a building, and ignoring direct sunlight, is the sum of three components: Externally Reflected light from objects on the outside of the building such as other buildings or trees, Internally Reflected light from surfaces within the building, and the Sky Component which refers to the light that reaches any point inside a building directely from the sky through an opening such as a window. Our experimental setup does not contemplate any outside objects, and does not contain any objects inside the test cube nor do we intend to take any characteristics form the building material in consideration for this analysis. So instead of calculating Daylight Factor or Daylighting we can concentrate on the Sky Component only. Calculations for this analysis were executed with the folowing parameters: Natural Light Levels, Over the Analysis Grid, Sky Iluminance calculated automatically from the model latitude, CIE Overcast Skycondition, Clean window and with the method of calculation set to the Regular Compliance Method.

SKY COMPONENT - 9122010 with shading device #1

Results from the Sky Component calculations in %:

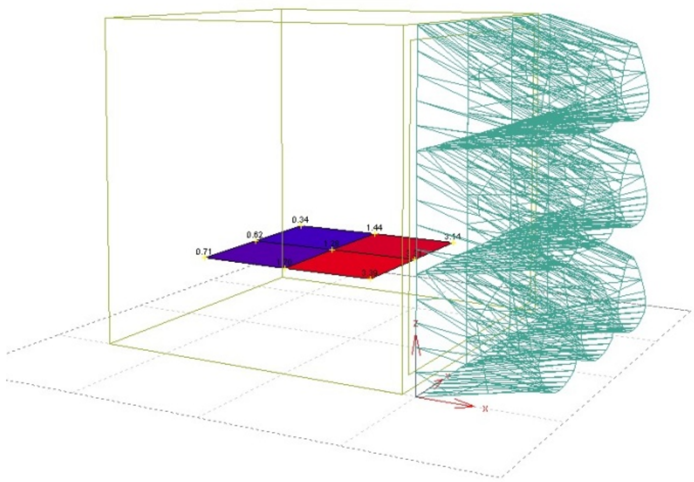
1.83044, 4.31323, 8.33599,
2.0599, 3.76528, 8.76392,
1.70947, 3.53217, 8.72027,



SKY COMPONENT - 9122010 with shading device #2

Results from the Sky Component calculations in %:

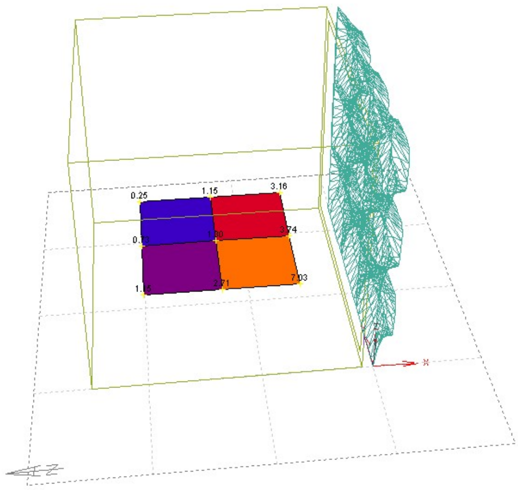
0.712085, 1.70468, 3.38817,
0.621702, 1.28123, 3.30745,
0.335263, 1.44399, 3.14138,



SKY COMPONENT - 9122010 with shading device #3

Results from the Sky Component calculations in %:

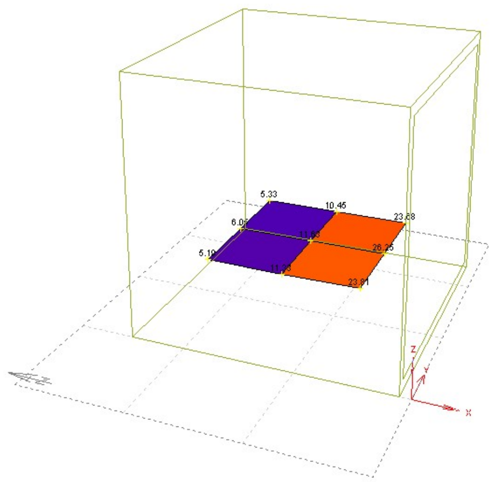
1.15439,	2.70751,	7.03321,
0.725018,	1.30338,	3.73682,
0.253527,	1.14791,	3.15983,



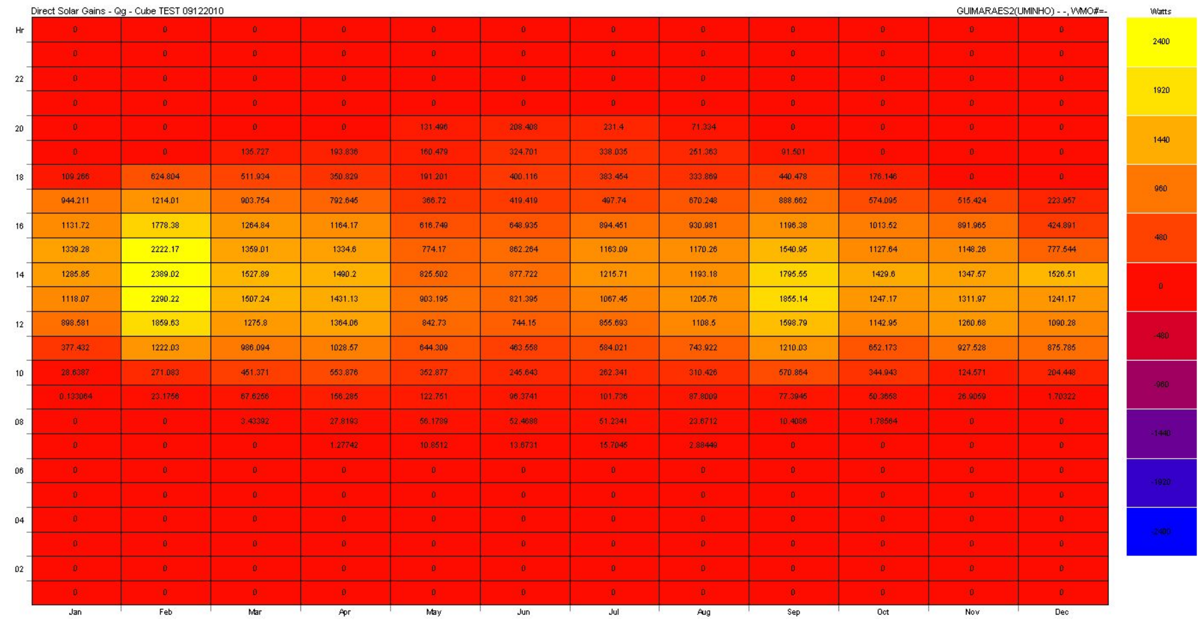
SKY COMPONENT - 9122010 without shading device

Results from the Sky Component calculations in %:

5.18769,	11.328,	23.8074,
6.05331,	11.6293,	26.2519,
5.32796,	10.4496,	23.6847,



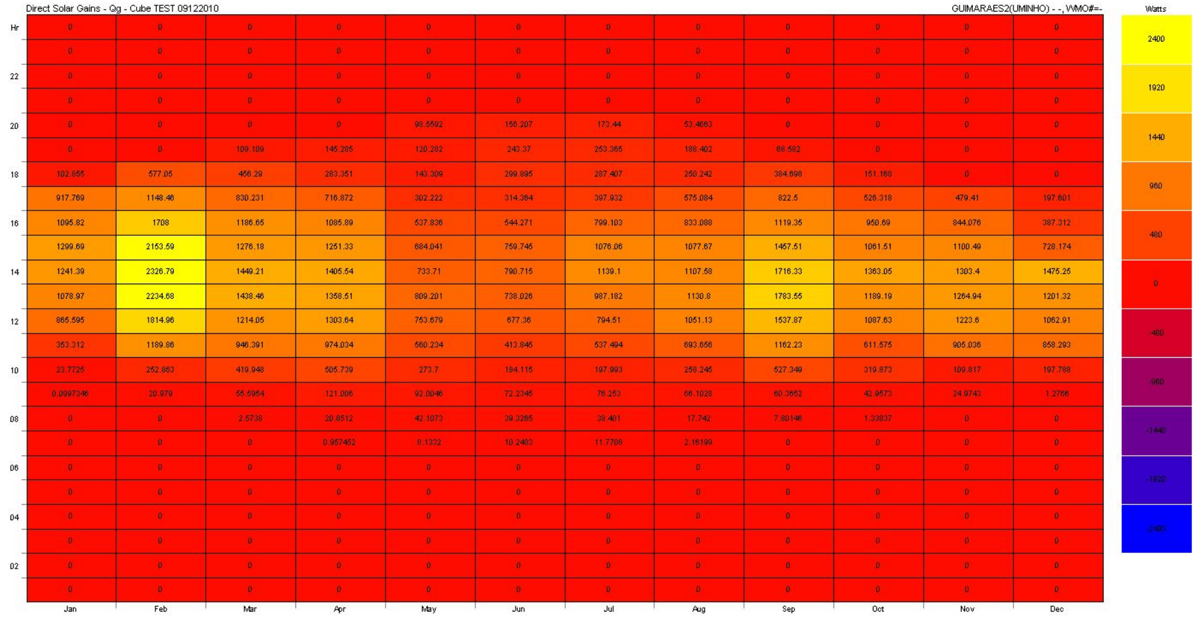
DIRECT SOLAR GAINS - 9122010 with shading device #1



ANNUAL LOADS TABLE - Direct Solar Gains – Qg - Cube TEST 09122010 - Monthly Averages

HOUR	JAN (Wh)	FEB (Wh)	MAR (Wh)	APR (Wh)	MAY (Wh)	JUN (Wh)	JUL (Wh)	AUG (Wh)	SEP (Wh)	OCT (Wh)	NOV (Wh)	DEC (Wh)
00	0	0	0	0	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0	0	0	0	0
02	0	0	0	0	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0	0	0	0	0
05	0	0	0	0	0	0	0	0	0	0	0	0
06	0	0	0	1	11	14	16	3	0	0	0	0
07	0	0	3	28	56	52	51	24	10	2	0	0
08	0	23	68	156	123	96	102	88	77	50	27	2
09	29	271	451	554	353	246	262	310	571	345	125	204
10	377	1222	986	1029	644	464	584	744	1210	652	928	876
11	899	1860	1276	1364	843	744	856	1108	1599	1143	1261	1090
12	1118	2290	1507	1431	903	821	1067	1206	1855	1247	1312	1241
13	1286	2389	1528	1490	826	878	1216	1193	1796	1430	1348	1527
14	1339	2222	1359	1335	774	862	1163	1170	1541	1128	1148	778
15	1132	1778	1265	1164	617	649	894	931	1196	1014	892	425
16	944	1214	904	793	367	419	498	670	889	574	515	224
17	109	625	512	351	191	400	383	334	440	176	0	0
18	0	0	136	194	160	325	338	251	92	0	0	0
19	0	0	0	0	131	208	231	71	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0

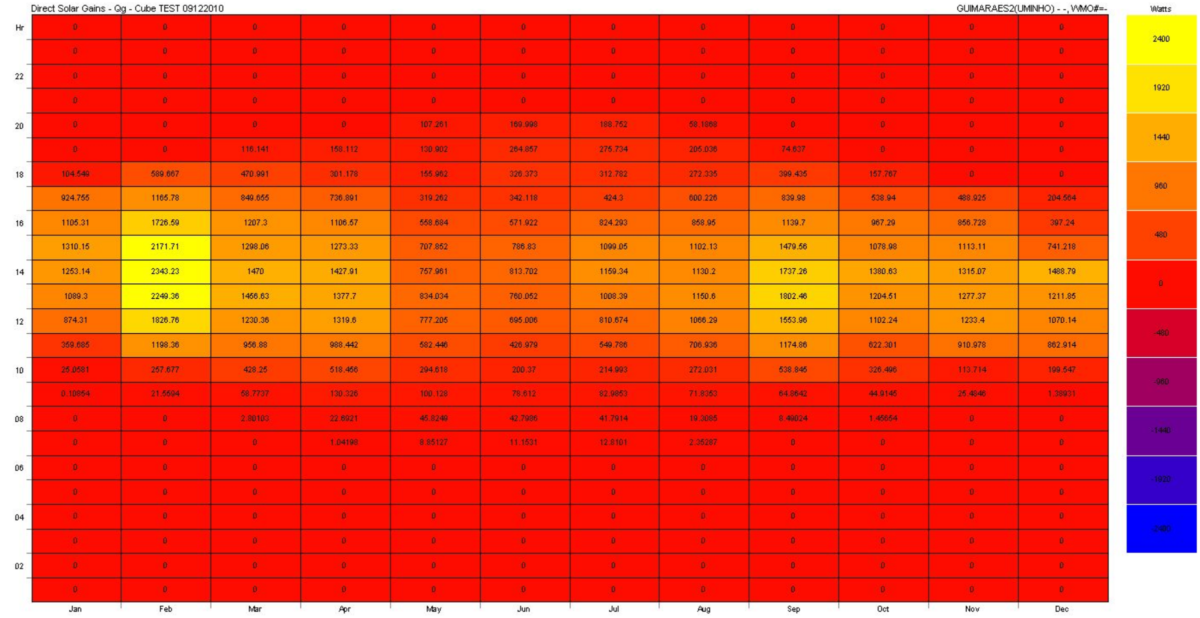
DIRECT SOLAR GAINS - 9122010 with shading device #2



ANNUAL LOADS TABLE - Direct Solar Gains – Qg - Cube TEST 09122010 - Monthly Averages

HOUR	JAN (Wh)	FEB (Wh)	MAR (Wh)	APR (Wh)	MAY (Wh)	JUN (Wh)	JUL (Wh)	AUG (Wh)	SEP (Wh)	OCT (Wh)	NOV (Wh)	DEC (Wh)
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
00	0	0	0	0	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0	0	0	0	0
02	0	0	0	0	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0	0	0	0	0
05	0	0	0	0	0	0	0	0	0	0	0	0
06	0	0	0	1	8	10	12	2	0	0	0	0
07	0	0	3	21	42	39	38	18	8	1	0	0
08	0	21	56	121	92	72	76	66	60	43	25	1
09	24	253	420	506	274	184	198	258	527	320	110	198
10	353	1190	946	974	560	414	537	694	1162	612	905	858
11	866	1815	1214	1304	754	677	795	1051	1538	1088	1224	1063
12	1079	2235	1438	1359	809	738	987	1131	1784	1189	1265	1201
13	1241	2327	1449	1406	734	791	1139	1108	1716	1363	1303	1475
14	1300	2154	1276	1251	684	760	1076	1078	1458	1062	1100	728
15	1096	1708	1187	1086	538	544	799	833	1119	951	844	387
16	918	1148	830	717	302	314	398	575	823	526	479	198
17	103	577	456	283	143	300	287	250	385	151	0	0
18	0	0	109	145	120	243	253	188	69	0	0	0
19	0	0	0	0	99	156	173	53	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0

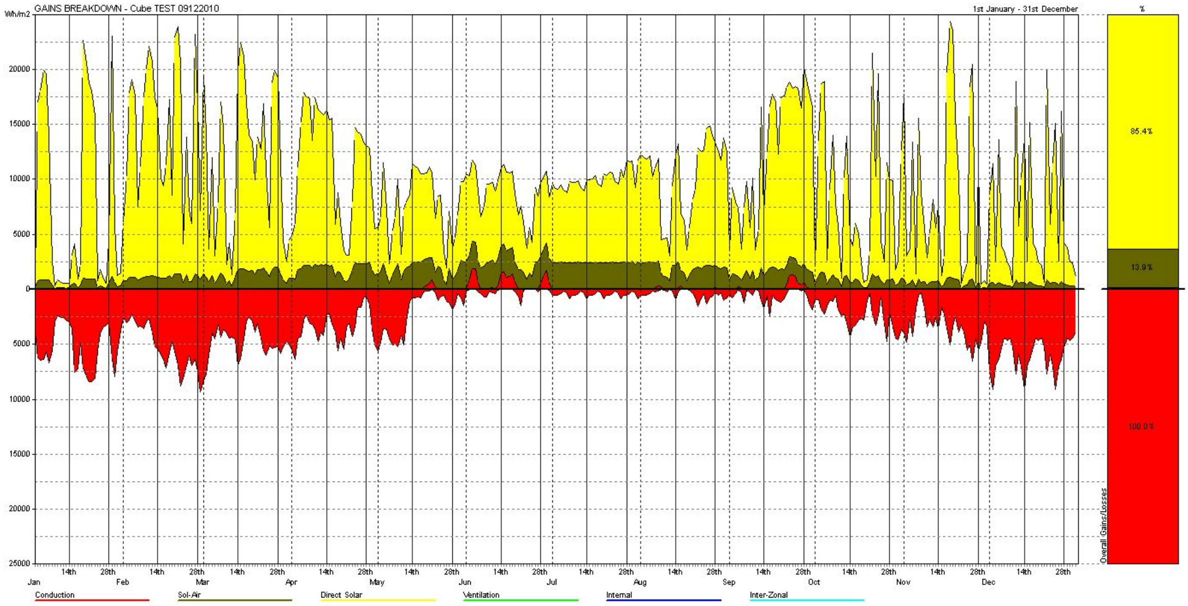
DIRECT SOLAR GAINS - 9122010 with shading device #3



ANNUAL LOADS TABLE - Direct Solar Gains – Qg - Cube TEST 09122010 - Monthly Averages

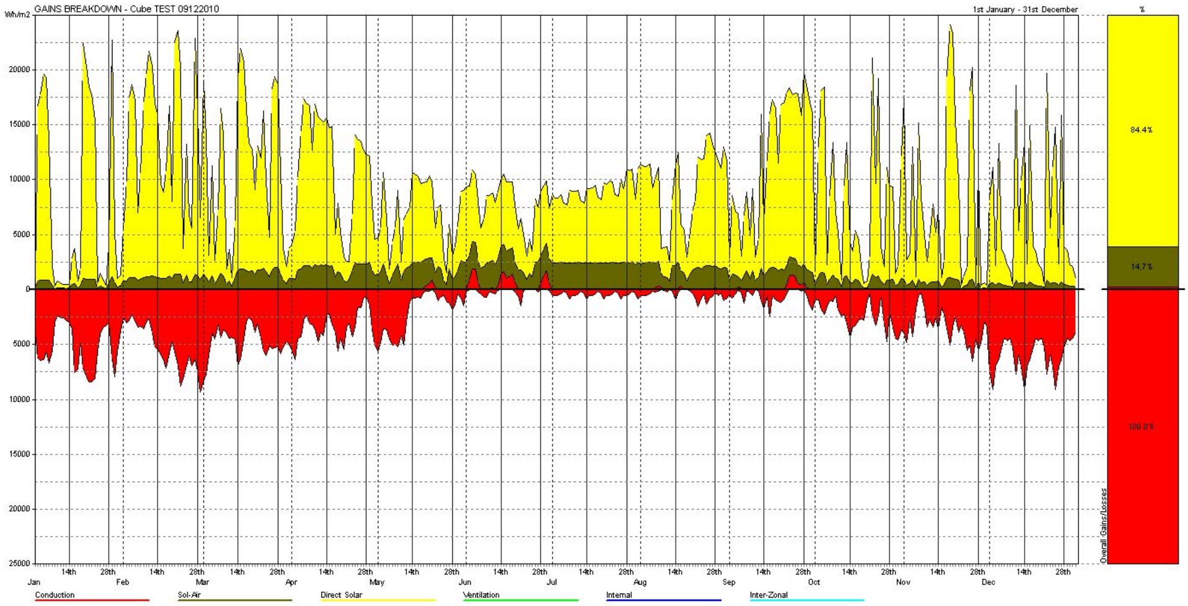
HOUR	JAN (Wh)	FEB (Wh)	MAR (Wh)	APR (Wh)	MAY (Wh)	JUN (Wh)	JUL (Wh)	AUG (Wh)	SEP (Wh)	OCT (Wh)	NOV (Wh)	DEC (Wh)
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
00	0	0	0	0	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0	0	0	0	0
02	0	0	0	0	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0	0	0	0	0
05	0	0	0	0	0	0	0	0	0	0	0	0
06	0	0	0	1	9	11	13	2	0	0	0	0
07	0	0	3	23	46	43	42	19	8	1	0	0
08	0	22	59	130	100	79	83	72	65	45	25	1
09	25	258	428	518	295	200	215	272	539	326	114	200
10	360	1198	957	988	582	427	550	707	1175	622	911	863
11	874	1827	1230	1320	777	695	811	1066	1554	1102	1233	1070
12	1089	2249	1457	1378	834	760	1008	1151	1802	1205	1277	1212
13	1253	2343	1470	1428	758	814	1159	1130	1737	1381	1315	1489
14	1310	2172	1298	1273	708	787	1099	1102	1480	1079	1113	741
15	1105	1727	1207	1107	559	572	824	859	1140	967	857	397
16	925	1166	850	737	319	342	424	600	840	539	489	205
17	105	590	471	301	156	326	313	272	399	158	0	0
18	0	0	116	158	131	265	276	205	75	0	0	0
19	0	0	0	0	107	170	189	58	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0

PASSIVE GAINS BREAKDOWN - 9122010 with shading device #1



CATEGORY	LOSSES	GAINS	CATEGORY	LOSSES	GAINS
FABRIC	100.0%	0.8%	SOL-AIR	0.0%	13.9%
SOLAR	0.0%	85.4%	VENTILATION	0.0%	0.0%
INTERNAL	0.0%	0.0%	INTER-ZONAL	0.0%	0.0%

PASSIVE GAINS BREAKDOWN - 9122010 with shading device #2

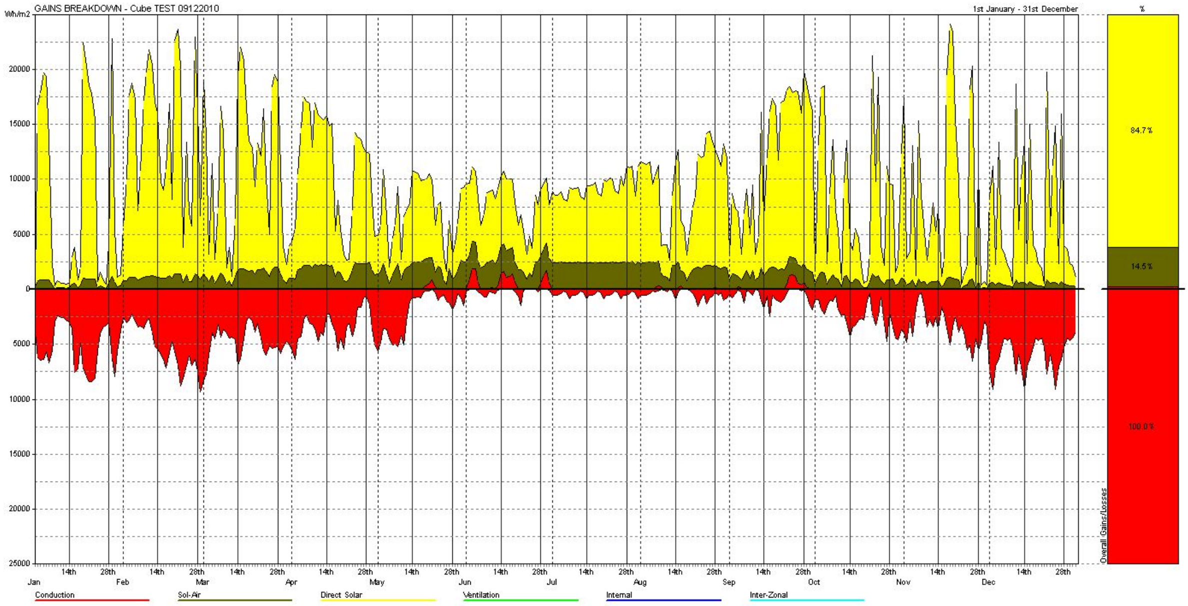


CATEGORY	LOSSES	GAINS	CATEGORY	LOSSES	GAINS
FABRIC	100.0%	0.9%	SOL-AIR	0.0%	15.4%
SOLAR	0.0%	83.7%	VENTILATION	0.0%	0.0%
INTERNAL	0.0%	0.0%	INTER-ZONAL	0.0%	0.0%

In order to get a quick verification of possible interference in the results of the small distance of 100mm at which the shading devices are placed in front of the cube, the same analysis procedure was rerun but this time with the shading device #2 fixed to the cube. No measurable difference can be noted.

CATEGORY	LOSSES	GAINS	CATEGORY	LOSSES	GAINS
FABRIC	100.0%	0.9%	SOL-AIR	0.0%	15.4%
SOLAR	0.0%	83.7%	VENTILATION	0.0%	0.0%
INTERNAL	0.0%	0.0%	INTER-ZONAL	0.0%	0.0%

PASSIVE GAINS BREAKDOWN - 9122010 with shading device #3



CATEGORY	LOSSES	GAINS	CATEGORY	LOSSES	GAINS
FABRIC	100.0%	0.8%	SOL-AIR	0.0%	14.5%
SOLAR	0.0%	84.7%	VENTILATION	0.0%	0.0%
INTERNAL	0.0%	0.0%	INTER-ZONAL	0.0%	0.0%

ANNEX C

The full set of 68 Pareto Optimum solutions from the second multi-objective optimisation.