

**Multidisciplinary design
optimisation as strategy
for building design**

S. Lamping

Multidisciplinary design optimisation as strategy for building design

by

Sander Lamping

in partial fulfilment of the requirements for the degree of

Master of Science

in Civil Engineering

at the Delft University of Technology.

An electronic version of this thesis is available at
<http://repository.tudelft.nl>.

Student:

Sander Lamping

Address: Mijnbouwplein 27
2628RT Delft
Student ID: 4014529
Email: sanderlamping@hotmail.com
Phone: +31 (0)6 39 86 10 87

Chairman:

Dr. ir. J. L. Coenders

Delft University of Technology
BEMNext Laboratory
White Lioness technologies

Email: j.l.coenders@tudelft.nl
Phone: +31 (0)20 737 19 97

Thesis committee:

Prof. dr. ir. J. G. Rots

Delft University of Technology

Email: j.g.rots@tudelft.nl
Phone: +31 (0)15 278 37 99

Dr. M. Turrin

Delft University of Technology

Email: m.turrin@tudelft.nl
Phone: +31 (0)6 29 32 18 39

Ir. C. J. Janssen

Delft University of Technology

Email: c.j.janssen@tudelft.nl
Phone: +31 (0)6 53 12 62 73

Daily supervisor:

Ir. S. Ren

Arup Amsterdam

Email: shibo.ren@arup.com
Phone: +31 (0)20 752 31 68

Preface

This Master's thesis was written as part of the Master curriculum of Building Engineering, with a specialisation in Structural Design. Building Engineering is a Master track of Civil Engineering at the Delft University of Technology. The research was carried out in collaboration with the Faculty of Civil Engineering and Geosciences and the engineering firm Arup, located in Amsterdam.

Firstly, I would like to thank the members of my graduation committee, dr. ir. Jeroen Coenders, prof. dr. ir. Jan Rots, dr. Michela Turrin, ir. Christien Janssen and ir. Shibo Ren, for their guidance and inspiration. Without their much appreciated support and advice, the outcome of this thesis would not have been the same.

Furthermore, I would like to thank the engineering firm Arup for providing me with a pleasant working environment and the resources needed to conduct this Master's thesis. I would also like to thank my colleagues at Arup for their interest in my research and for offering help when needed.

Lastly, on a more personal note, I would like to express my gratitude to my family and friends. Specifically to my parents Rob and Sylvia, my brother Mischa and my girlfriend Naomi, for their motivation and support.

Sander Lamping
Delft, June 2016

Abstract

When designing a building, numerous different disciplines are involved. With the complexity of building design problems growing, almost each of these disciplines requires their own experts. These experts work together, although in sequence, on one and the same building design. Most specialists only have limited knowledge about the other disciplines, which means each of them are (almost) exclusively focused on the performance of their own field of expertise (Uijtenhaak and Coenders, 2012). However, in the final building different conflicting disciplines should form an integrated design. Because of the specific knowledge per expert and the growing size of the design team, decisions are difficult to make and can take a significant amount of time. This results in only few design alternatives considered during the (time limited) conceptual design stage. However, decisions made in this stage have a substantial impact on the final performance of a building. Therefore, it is important to reduce the iteration time of a design cycle, allowing design teams to consider more design alternatives within the same amount of limited time. Many potential alternatives are desirable, to be able to compare them with each other and to understand what influence is caused by which design choice (Uijtenhaak and Coenders, 2012). This can lead towards better informed decision making and potentially towards better performing and more integrated building designs.

Multidisciplinary design optimisation could be able to overcome the aforementioned challenges and could lead towards a more integrated design process. MDO allows designers to include different disciplines in the optimisation process simultaneously. This ensures potential conflicts between disciplines are already taken into account during the optimisation. Consequently it can reduce extra design iterations in later stages of the process (Yang et al., 2015). The optimal solution of the joined problem can be superior to the design which is formed by optimising the different disciplines sequentially (Martins and Lambe, 2013). Multidisciplinary design optimisation techniques are already applied in several fields, such as aerospace and automotive engineering. Nonetheless, their application in the architecture, engineering and construction industry has been modest (Flager et al., 2009). Earlier research (Flager and Haymaker, 2007; Geyer, 2009; Gerber and Lin, 2014; Yang et al., 2015; Jansen et al., 2014) however showed that MDO approaches could be beneficial in the AEC industry as well.

In this research project, an approach is proposed for applying multidisciplinary design optimisation on building design problems. The software used in the approach are Rhinoceros and Grasshopper for the parametric configuration of the geometry, the plug-in Octopus driving the optimisation process with its multi-objective genetic algorithm and a custom post-processing tool PPOR for comparing and filtering different results and investigating the trade-offs between different objectives. To test the functionality of the proposed approach, it was applied on a rel-

evant case study. The large span steel roof of the Ukrainian Donbass Arena was chosen as test case. Focus was given on the structural and acoustic performance of the roof. The structural performance was described by the structural mass of the roof, whereas the acoustic deviation throughout the stadium was measured for the acoustic assessment. The plug-in Karamba was used for the structural analysis and a custom ray-tracing tool, SPL-Estimator, was developed for the acoustic analysis. A parametric model of the football stadium was set up, in which the geometry of the roof was described by 11 variables.

The multidisciplinary design optimisation approach was able to find solutions which improved the baseline design of the arena up to 58.3% for the structural objective and 22.8% for the acoustic objective. In 32 hours, the optimisation algorithm was able to find 72 solutions that outperformed the baseline design for both objectives. Next to the mentioned improvements in performance, the approach provides the user with insight in trade-offs between the different objectives and the influence of different design variables. The proposed approach proved to be applicable for different design problems, with alterations for the model set up, analysis methodology and optimisation organisation required according the design problem. Furthermore, the approach is usable for architects and engineers with knowledge of Rhinoceros and Grasshopper. However, the computation time needed to evaluate different solutions showed to be excessive, which is a major drawback for applying the proposed approach. Further research on improving the calculation speed of the approach is therefore recommended.

Contents

Preface	vii
Abstract	ix
Acronyms	xiii
1 Introduction	15
1.1 Problem definition	17
1.2 Research objectives	17
1.3 Research questions	18
1.4 Domain and scope	19
1.5 Thesis outline	19
2 Multidisciplinary design optimisation	21
2.1 Design optimisation	22
2.2 MDO architectures	24
2.3 Optimisation algorithms	26
2.4 Multidisciplinary design optimisation in the AEC industry	28
3 Acoustic and structural analysis	33
3.1 Acoustic analysis	33
3.2 Structural analysis	43
4 Optimisation approach and workflow	51
4.1 Octopus	51
4.2 MDO approach and process	53
4.3 SPEA-2 and HypE	56

Contents

4.4	Post processing	57
4.5	Further possible applications	62
5	Case study	65
5.1	Design problem	66
5.2	Modelling	73
5.3	Baseline design	83
6	Optimisation results	85
6.1	Optimal solutions	85
6.2	Processing the results	89
7	Validation	93
8	Discussion	103
8.1	Optimisation approach	103
8.2	Case study	105
9	Conclusions	107
10	Recommendations	109
	Bibliography	111
	Glossary	115
A	Custom ray tracing tool (SPL-Estimator)	117
B	Structural benchmarking calculations	123
B.1	Cantilevers - case 1 and 2	124
B.2	Simply supported - case 3 and 4	128
C	Post processing tool (PPOR)	131
D	List of Figures	135
E	List of Tables	139

Acronyms

AEC	Architecture, Engineering and Construction
BEM	Built Environment Modelling
BEMNext	The next generation of Built Environment Modelling
BIM	Building Information Modelling
CAD	Computer Aided Design
CAE	Computer Aided Engineering
DSM	Design Structure Matrix
FEA	Finite Element Analysis
FEM	Finite Element Method
GA	Genetic Algorithm
HypE	Hypervolume Estimation
MDO	Multidisciplinary Design Optimisation
MOGA	Multi-objective Genetic Algorithm
MOO	Multi-objective Optimisation
NURBS	Non-uniform Rational Basis Spline
SLS	Serviceability Limit State
SOO	Single-objective Optimisation
SPEA	Strength Pareto Evolutionary Algorithm
SPL	Sound Pressure Level
SWL	Sound Power Level
ULS	Ultimate Limit State

1

Introduction

When designing a building, numerous different disciplines are involved. Architecture, structural design and building services are just a few examples of disciplines which need to be considered. With the complexity of building design problems growing, almost each of these disciplines requires their own experts. These experts work together, although in sequence, on one and the same building design (Uijtenhaak and Coenders, 2012). Because of the specific knowledge per expert and the growing size of the design team, decisions are difficult to make and can take a significant amount of time. On top of that, the performance of different disciplines often conflict with the performances of others. Most specialists only have limited knowledge about the other disciplines, which means each of them are (almost) exclusively focused on the performance of their own field of expertise (Uijtenhaak and Coenders, 2012). However, in the final building different disciplines should form an integrated design. Integrating different disciplines again adds valuable time required to coordinate the building design process (Flager et al., 2009).

This challenging and lingering design process results in only few design alternatives considered during the (time limited) conceptual design stage. During this phase, architects and engineers will collectively define their objectives and design different alternatives. These will then be evaluated to find solutions which meet the objectives best. This process is however usually performed per discipline, which means the coordination of it is very labour intensive. Consultation is needed to ensure the different disciplines are consistent, so that the conceptual design process is more often focused on resolving conflicts between the different disciplines instead of optimising the performance of the building as a whole (Flager and Haymaker, 2007).

Flager and Haymaker (2007) show a study of a leading firm, which states it takes architects and engineers over a month to complete the aforementioned design cycle. This means that they average less than three design cycles per project. Consequently, design decisions are made

1. Introduction

with relatively little information about the performances of the chosen design compared to possible alternatives.

However, decisions made in the conceptual design stage have a substantial impact on the final performance of a building. Therefore, it is important to reduce the iteration time of a design cycle, allowing design teams to consider more design alternatives within the same amount of limited time. Many potential alternatives are desirable, to be able to compare them with each other and to understand what influence is caused by which design choice (Uijtenhaak and Coenders, 2012). This can potentially lead towards better performing and more integrated building designs.

Improvements in the design process can already be seen in the last couple of years. The development of Building Information Modelling and computational analysis approaches permit different disciplines to model and simulate building performances in a virtual environment. On top of that, BIM helps to manage important information between both disciplines and design stages. These tools are however more intended to analyse design alternatives and manage design information, rather than to help explore the design space (Flager and Haymaker, 2007).

Parametric design, associative design and design optimisation (see Glossary), in contrast, are able to help exploring the design space and are becoming common terms in the current building atmosphere. Parametric and associative modelling (see Glossary) simplify the process of changing building designs (Uijtenhaak and Coenders, 2012), by allowing the rapid change of both geometric and non-geometric variables according to particular design logic (Flager et al., 2009). This makes it easier to quickly 'generate' different design alternatives. By generating as much as possible alternatives, the knowledge about the design is enlarged (Uijtenhaak and Coenders, 2012).

When the design requirements are expressed in measurable performance indicators, the parameters of the building model can be altered in such a way an alternative will satisfy these performance requirements best. By letting a computer generate and analyse design alternatives, the design space can be explored and performances measured. Optimisation algorithms are able to guide this investigation towards the best performing designs. Design optimisation is originally carried out with respect to one governing objective, such as cost or weight and is done by a function which represents the performance and variables covering the design space (Uijtenhaak and Coenders, 2012).

Optimisation processes are still mainly focused on isolated disciplines. This means that the final multidisciplinary design is assembled out of separate 'optimal' solutions added together, instead of an overall optimal solution. Besides that, the trade-off between multiple disciplinary aspects is often unknown to the different disciplinary parties. It is highly contextual and is different for each project and for each discipline (Jansen et al., 2014). Although the multidisciplinary design might consist out of 'joined' optimal solutions, it is not an optimal one itself.

Multidisciplinary design optimisation can be the next step towards a more integrated design process. MDO allows designers to include different disciplines in the optimisation process simultaneously. This ensures potential conflicts between disciplines are already taken into account during the optimisation. The optimal solution of the joined problem can be superior to the design which is formed by optimising the different disciplines sequentially (Martins and Lambe, 2013).

1.1 Problem definition

Multidisciplinary design optimisation has not often been applied on 'real life' building design problems. Although the examined problems were real-size, the solutions were highly simplified and limited. MDO has however shown potential in other fields of engineering and may thus be of value to overcome some of the difficulties faced during the building design process.

There are however several reasons why multidisciplinary design optimisation is not regularly used yet in the building design process. Most of these reasons arise from the existing differences between the AEC industry and engineering fields as aerospace and automotive engineering. Consequence of these differences is that existing MDO approaches used in other engineering fields might not be suitable to apply on building design problems.

First of all, performances of several disciplines in a building design are often less well defined than in other fields of engineering. This is due to the importance of different qualitative aspects of a building design, such as aesthetics, which are of great importance for the design 'performance' but subjective and difficult (or impossible) to quantify. If it is not possible to quantify an aspect numerically, the optimisation algorithm is not capable of improving it. To apply multidisciplinary design optimisation in the AEC industry, a technique of dealing with these aspects is needed (Geyer and Beucke, 2010).

Furthermore, building designs are often one of a kind which makes standardisation of an applicable MDO approach difficult. For different designs, different disciplines are decisive. Also the interactions between different disciplines are different per building and much less 'organised' than in designs in other engineering fields. This affects the applicability of a system to support the design process for different buildings (Jansen et al., 2014).

Finally, building design problems usually require many parameters to evaluate the performances of the different disciplines, which complicates automated optimisation in an acceptable amount of time (Jansen et al., 2014).

1.2 Research objectives

The goal of this research is to overcome these mentioned difficulties and to apply multidisciplinary design optimisation on a building design problem. The steel roof of the Donbass Arena in Ukraine (see Chapter 5) will be used as a case study. Stadium roofs are large building structures where multidisciplinary design is of growing complexity and importance. Stadiums are nowadays often used for events other than sports. On top of that, capacity and thus the size of stadiums are increasing. This all leads to more complex design problems, for which multidisciplinary design optimisation could be of great help. Note that the case study will focus on the structural and acoustic performance of the Donbass Arena roof. This choice will be further explained in Section 1.4.

1. Introduction

The main objective of this research can be formulated as follows:

Find a suitable approach (1) to apply multidisciplinary design optimisation (acoustic and structural optimisation) on a relevant building design problem (steel stadium roof) to investigate the potential (2) of multidisciplinary design optimisation in the AEC industry.

Figure 1.1 shows a flowchart of the foreseen methodology of this research. The first step is to investigate MDO more elaborately, after which an approach is chosen for applying MDO on the case study. After validating this approach, it is used to optimise the steel roof of the Donbass Arena. Finally, it will be investigated if this approach is applicable on other design problems.



Figure 1.1: Flowchart illustrating the foreseen research methodology

1.3 Research questions

1.3.1 Main research question

The main research question, which follows from the previously defined objectives, is as follows:

'In what manner and to what extent could a multidisciplinary design optimisation approach be applicable and beneficial for a large span steel stadium roof design, focused on its structural and acoustic quality?'

1.3.2 Sub research question

The following sub research questions focus on the different aspects of the main research question.

- What are the potential benefits of a multidisciplinary design optimisation approach in the AEC industry and what is needed for such an approach to be applicable? (Chapter 2)
- In what manner could multidisciplinary design optimisation be applied on the aforementioned building design problem? (Chapter 3, Chapter 4, Chapter 5)
- Is the proposed approach applicable on other (building) design problems? (Chapter 4, Chapter 8)

1.4 Domain and scope

As it is difficult to assess all different disciplines involved in this research on multidisciplinary design optimisation, it was chosen to focus on only two disciplines in the case study design problem. The focus will mainly be on the disciplines structure and acoustics. The size alone already makes the structure of a stadium roof interesting and optimising of the structure can lead to a significant amount of material savings. Acoustics are becoming more and more important in the design of stadium, as they are nowadays also often used for concerts. The size of stadiums however makes it difficult to achieve the desired acoustic quality. Besides, acoustics are important for the atmosphere in the stadium during sporting events and the environment of the building may not have too much nuisance of the noise. Both the structure and the acoustic quality of a stadium strongly depend on the geometry of the stadium and its roof, which may lead to potential conflicting objectives.

Furthermore, it was chosen to focus this research on an multidisciplinary design optimisation approach within the applications Rhinoceros (R. McNeel and Associates, 2016) and its plug-in Grasshopper (Davidson, 2016), as these applications are relatively known by architects and engineers. Section 2.4 elaborates more on this delimitation.

1.5 Thesis outline

The main report consists ten chapters. In this section, the content of the following nine chapters will be shortly described.

Chapter 2 will introduce and explain some key elements of multidisciplinary design optimisation.

Chapter 3 focuses on the analysis methods which are used to assess both the acoustic and structural performance in the case study. These methods will be exemplified and validated.

Chapter 4 will focus on the chosen approach and workflow to apply MDO on the described case study. Different set-ups are tested and the different steps of the workflow are explained.

Chapter 5 illustrates the chosen case study and explains the objectives, variables and constraints of the optimisation problem. It will furthermore focus on the modelling of the case study.

Chapter 6 will show the found optimisation results and will further elucidate these results.

Chapter 7 focuses on the validation of the results and findings of the optimisation process.

In Chapter 8, the results and findings of the research will be discussed.

Chapter 9 will show the conclusions of this research.

Chapter 10 will suggest possibilities for further research.

2

Multidisciplinary design optimisation

Multidisciplinary design optimisation (MDO) is an approach in engineering, which employs optimisation methods to solve or help solve design problems including several different disciplines or subsystems (Martins and Lambe, 2013). MDO can be traced back to the 1960's, when the experience in structural optimisation was extended to include other disciplines (Schmit, 1960). The first applications were focused on the aerodynamics and structure of aircraft wing design (Martins and Lambe, 2013). Multidisciplinary design optimisation allows designers and engineers to optimise different disciplines simultaneously, instead of sequentially or independently. As a result, the performance of the multidisciplinary system is not only determined by the performances of the individual disciplines, but by the interactions between these disciplines as well. Consequently it can reduce extra design iterations in later stages of the process (Yang et al., 2015). It can lead towards better performing solutions than independent and sequential approaches.

By solving MDO problems at the beginning of the design process and benefiting advanced computational analysis tools, designers are able to both improve the performance of a design and reduce the time and cost of the design cycle (Martins and Lambe, 2013). However, an MDO approach also considerably increases the complexity of the design problem and thus the time and effort needed to solve it (Yang et al., 2015). Multidisciplinary design optimisation techniques are already applied in several fields, such as aerospace and automotive engineering. Nonetheless, their application in the architecture, engineering and construction industry has been modest (Flager et al., 2009). Earlier research (Flager and Haymaker, 2007; Geyer, 2009; Gerber and Lin, 2014; Yang et al., 2015; Jansen et al., 2014) showed MDO approaches could however be beneficial in the AEC industry as well.

In this chapter, the concept of multidisciplinary design optimisation and its potential and applicability in the AEC industry is further described.

2. Multidisciplinary design optimisation

2.1 Design optimisation

Design optimisation is the process of finding the best performing design by altering its parameters, with regard to chosen objectives and under specified constraints (MathWorks, 2016a). An optimisation problem can be described as in Equation 2.1 (Boyd and Vandenberghe, 2004).

$$\begin{array}{ll} \text{minimise} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \end{array} \quad (2.1)$$

Where $f(x)$ is the objective function to be minimised over the variable x . The objective function usually describes the performance of a particular aspect of the design. Functions $g_i(x)$ are the specified constraints, which should be met. An example of the objective function could be the weight of the structure of the building with the maximum allowable deflection of the structure as constraint. The formulation of the objectives and constraints is a key aspect for the optimisation process as it will significantly influence the outcomes. Note that a problem could also contain multiple objectives.

By exploiting the calculation power of a computer, a large number of design alternatives can be evaluated; many more than would be possible with manual trial-and-error methods (Yang et al., 2015). Exploring the design space elaborately also supports with decision making (Bradner and Davis, 2013). Computational optimisation generally involves three main components. A mathematical model which describes the design, a simulator which analyses each alternative and an optimiser which guides the search process (Yang et al., 2015; Jansen et al., 2014).

2.1.1 Single-objective optimisation

Design optimisation can be divided in two groups based on its number of objectives. Single-objective optimisation is the simplest one (Yang et al., 2015), at which the optimisation algorithm searches for the best performing solution with regard to the one objective. For many engineering (and real-life) problems however, objectives conflict with each other and optimisation towards a single objective could lead to unacceptable results with respect to other objectives (Konak et al., 2006).

To deal with multiple objectives (possibly from different disciplines) with single-objective optimisation, a sequential process is required. Different objectives need to be optimised in isolation and in a predetermined order, with the other design variables remaining fixed. This can cause a long design cycle due to impossibility of simultaneous optimisation and sub-optimal results due to a reducing degree of freedom further in the process (Yang et al., 2015).

Other approaches to deal with multiple objectives in S.O.O. are to combine the different objective in one composite function or move all except one function to the constraints. The problem of such approaches is the arbitrary and subjective manner of deciding the scaling weights for combining the functions or constraining values in case of 'constraining objectives', which has a large impact on the outcome of the optimisation process (Konak et al., 2006).

2.1.2 Multi-objective optimisation

Multi-objective optimisation, in contrast, allows to include more than one objective function to be optimised simultaneously. Consequently, it is able to overcome the aforementioned limitations of single-objective optimisation (Yang et al., 2015). As objectives of a problem generally conflict with each other, multi-objective optimisation searches for the best possible trade-offs between the different objectives. The result is an optimal 'non-dominated' set of solutions, instead of a single optimal solution as found when applying single-objective optimisation (Konak et al., 2006).

This 'non-dominated' set of solutions is known as a Pareto front. Dominance can be explained as follows (Echenagucia et al., 2014):

- If solution A outperforms solution B in at least one objective function and equals or outperforms solution B in all other functions, then solution A dominates solution B.
- If solution A equals or outperforms solution B in all objective functions except one in which solution B outperforms solution A, then solution A and B do not dominate each other.

Consequently, Pareto optimality means that it is not possible to improve a single objective without worsening at least one other. An example of a Pareto front (orange solutions) is shown in Figure 2.1. Each of the points represents a solution to the minimisation problem.

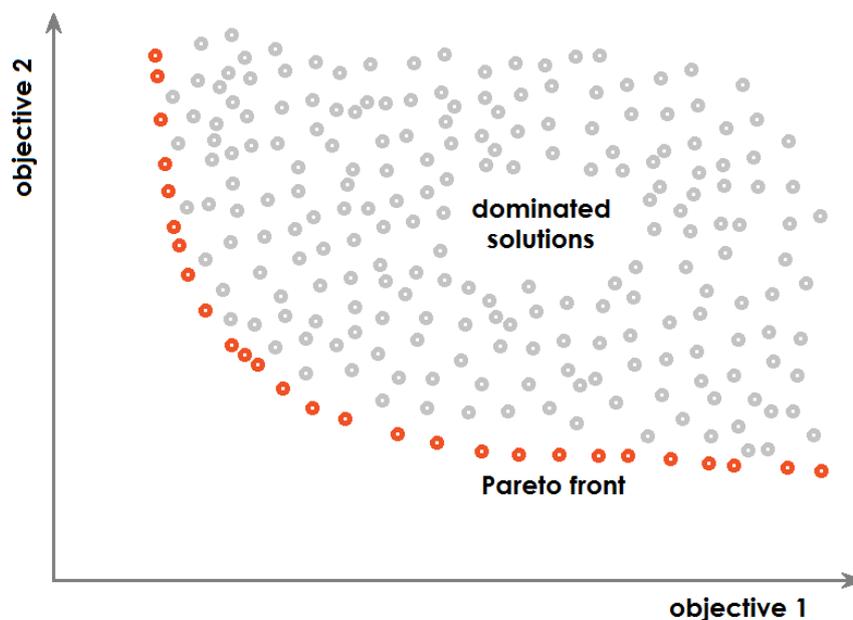


Figure 2.1: Example of a Pareto front

2. Multidisciplinary design optimisation

Generally, it is difficult and at times almost impossible to solve multidisciplinary design problems by looking at a single objective function or performance indicator. That is why in most cases multidisciplinary design optimisation is multi-objective as well (Echenagucia et al., 2014).

Multi-objective optimisation not only helps to find 'optimal' solutions, but also gives architects and engineers more understanding about the trade-offs between the different objectives. This is not necessarily the case when different performance values are combined in a single objective. It supports designers to efficiently explore the design space and helps them to make well founded decisions (Ren et al., 2011).

2.2 MDO architectures

An important consideration when implementing multidisciplinary design optimisation is the organisation of the several discipline analysis models, approximation models and optimisation software in combination with the problem formulation. This combination is commonly referred to as MDO architecture. The choice of architecture, together with the chosen optimisation algorithm, can have a major influence on the computation time and the final solution (Martins and Lambe, 2013).

The MDO architecture describes (1) how the different disciplines (or systems) are coupled and (2) how the overall optimisation is solved. It can be monolithic or distributed. In monolithic approaches, a single optimisation problem is solved. Distributed approaches divide the problem in several 'sub'-problems, each with a small set of the variables and constraints (Martins and Lambe, 2013). This can for example be of use when parallel computing is desirable. The choice for the most appropriate MDO architecture is dependent on the human and computational organisation, the available algorithms and the design problem itself (Martins and Lambe, 2013).

A monolithic approach is more simple than a distributed approach. The values of the 'coupled' variables stay coupled and are determined by the multidisciplinary analysis. This makes the implementation relatively easy. The modularity of a monolithic approach is however poor, which means it brings difficulties to parallel computing and is less compatible with the organisational structure of an interdisciplinary design team (Yang et al., 2015). In a distributed approach the 'coupled' variables are disconnected and determined by the system level optimiser. A distributed architecture has the drawback of sometimes permitting the exploration of infeasible regions of the design space. This leads towards an increased problem size and the risk of ending the process at an infeasible 'optimal' design (Martins and Lambe, 2013).

An example of a mathematical description of a basic single-level MDO approach (so called multidisciplinary feasible architecture (Martins and Lambe, 2013)) is assumed in Equation 2.2 (Yang et al., 2015). According to this system some key terms are exemplified.

2. Multidisciplinary design optimisation

$$\begin{aligned}
 & \text{minimise} && f_1(x_1, z_{sh}, z_1, y_{21}) \\
 & && f_2(x_2, z_{sh}, z_2, y_{12}) \\
 & && f_3(x_3, z_{sh}, z_3) \\
 & \text{with respect to} && z_{sh}, z_1, z_2, z_3 \\
 & \text{subject to} && g(x_1, z_{sh}, z_1) = 0 \\
 & && h(x_2, y_{12}, z_{sh}, z_2) \leq 0
 \end{aligned} \tag{2.2}$$

In this description z_i are the design variables. Subscript $_{sh}$ represents (global) design variables which are shared between different subsystems, whereas subscripts $_{1,2,3}$ indicate the (local) design variables which are specific to a single subsystem (Yang et al., 2015). These design variables are always under explicit control of the optimiser (Martins and Lambe, 2013). The used subscripts are also applied for describing the other variables.

State (or disciplinary) variables are indicated with x_i . These variable are able to vary during the disciplinary analysis to find an equilibrium in the state equations. A disciplinary analysis is a simulation which models the behaviour of one aspect of the multidisciplinary system and consists solving (a system of) equations. Note that state variables may or may not be controlled by the optimiser, depending on the selected problem formulation (Martins and Lambe, 2013). In the given example they are not controlled by the optimiser.

The coupling variables, specified with y_{ij} , model the interactions between different subsystems. The double indexation indicates that the variable is transmitted from the i^{th} subsystem to the j^{th} subsystem (Yang et al., 2015).

Similar to the earlier described optimisation problem, functions f_i are objective functions and functions g and h are the design constraints. In Figure 2.2, a schematisation of the described system is shown.

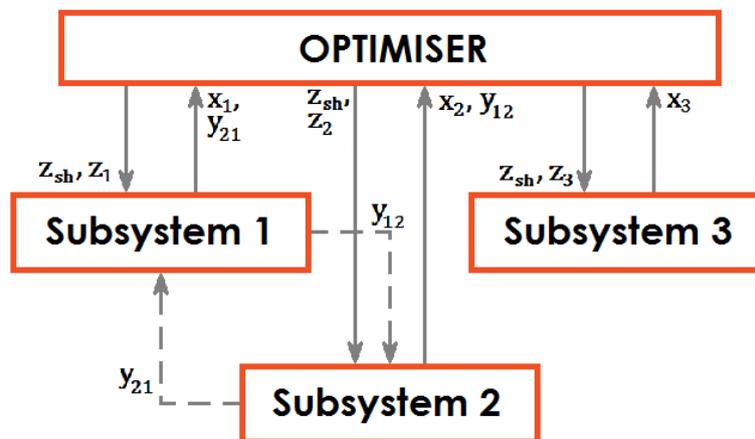


Figure 2.2: Schematisation of described multidisciplinary system

2. Multidisciplinary design optimisation

When a distributed MDO architecture is applied, three major steps (Ren et al., 2011) can be distinguished. These will be elaborated below.

- The first step is to analyse how to decompose the large design problem into individual elements. This decomposition can be done per discipline or building component, depending on the optimisation problem. However, a decomposition per discipline seems most appropriate in the AEC industry, as disciplines are often assessed by different design teams.
- When a decomposition by discipline is chosen, the next step is to classify the different sorts of variables. It should be outlined, which variables are important for several disciplines and which ones only influence a single discipline.
- Lastly, the optimisation process itself can start. The design per discipline is realised at subsystem level and the conflicts between the different disciplines are synchronised by an optimiser at system level. This optimiser is also responsible for optimising towards the overall design objectives.

As design problems faced in the conceptual design process in the AEC industry (in contrast with most problems faced in aerospace) generally do not have strong couplings (Jansen et al., 2014), a complex distributed MDO architecture might not always be required. When design problems are however complex and disciplines strongly coupled, neglecting the couplings may lead to oversimplification and underestimation of the problem (Yang et al., 2015). That is why a good examination of the multidisciplinary design problem is important prior to determining an appropriate MDO architecture.

2.3 Optimisation algorithms

Various different architectures can be chosen to solve a given optimisation problem, but at least as many algorithms could be chosen. Optimisation algorithms are sets of step-by-step operations (Wikipedia, 2016a) which drive the search for the minimum or maximum of the objective function. Just as the choice of architecture, the choice for a suitable algorithm depends on the design problem and has a large influence on the final design (Martins and Lambe, 2013). Different sorts of algorithms have their own positive and negative points.

This paragraph will mainly focus on multi-objective optimisation algorithms, as a multi-objective approach is a more appropriate for multidisciplinary design optimisation than a single-objective approach. The goal of a multi-objective optimisation algorithms is to find the entire Pareto set; this is however impossible. Due to the size of the set and the computational infeasibility of proving the optimality of a solution, a more practical approach is to investigate a set of solutions which represents the Pareto set as well as possible (Konak et al., 2006). Consequently, a multi-objective optimisation approach should achieve the following goals (Zitzler et al., 2000):

2. Multidisciplinary design optimisation

- The best-known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set is a subset of the Pareto optimal set.
- Solutions in this best-known Pareto set should be uniformly distributed across the Pareto front to accurately show the trade-offs between different objectives.
- The best-known Pareto front should cover the entire spectrum of the actual Pareto front; also at the extremes of the objective function space.

This research will focus on evolutionary algorithms and genetic algorithms in particular. The concept of these algorithms can be traced back to the 1960's and 1970's and are inspired by Charles Darwin's theory of evolution (Konak et al., 2006). According to survival of the fittest, strong individuals have a larger chance of passing their genes to future generations by reproduction than weaker individuals. Together with the chance of random positive modifications in genes, this process leads to a stronger population (Konak et al., 2006).

Just as other algorithms, evolutionary algorithms has positive aspects but limitations and drawbacks as well. First of all, evolutionary algorithms require many iterations. Secondly, an optimal solution is not guaranteed. The process will run indefinitely and the final solution is not proven to be the problem's optimum (Rutten, 2010).

In contrast, the benefits make evolutionary interesting for applying them for multi-objective optimisation problems. They are able to deal with a wide range of problems, from problems which are poorly formulated to problems with discrete design variables. Because of their global approach, they have the potential of finding global optimal solutions. Their biggest advantage in this case is however their ability to keep finding solutions. Generally, each generation of solutions is better than its predecessor. As a result, when the process is stopped prematurely, the result is always a solution. Solutions might not be close to the optimal yet, but superior to the initial values (Rutten, 2010).

In genetic algorithms, a solution vector is called a chromosome. A chromosome consists out of different discrete units called genes. These genes are the design variables of the model. As the genes change the state of the model, the fitness of the model changes. The 'fitness landscape' with two gene is shown in (1) in Figure 2.3 (Rutten, 2010). Genetic algorithms work with a collection of chromosomes, called a population. This population is generally initialised randomly (2). The fitness of each chromosome is evaluated and the fittest chromosomes are kept for breeding (3). The next generation consists of this breed and, if the process is going as intended, is fitter than the previous generation (4). This process is repeated (5) and eventually the search converges towards the optimal solution(s) (Rutten, 2010; Konak et al., 2006).

Genetic algorithms use two kind of operators to generate new solutions from existing ones, namely crossover and mutation. In crossover two chromosomes are chosen (parents) and combined to form a new chromosome (offspring). The parents are selected from the existing population, with a preference for fitness. Mutation introduces random changes in the characteristics of the chromosomes and is typically applied at gene level. Whereas crossover drives the population to converge, mutation reintroduces diversity to help the search process to escape from local optima (Konak et al., 2006).

2. Multidisciplinary design optimisation

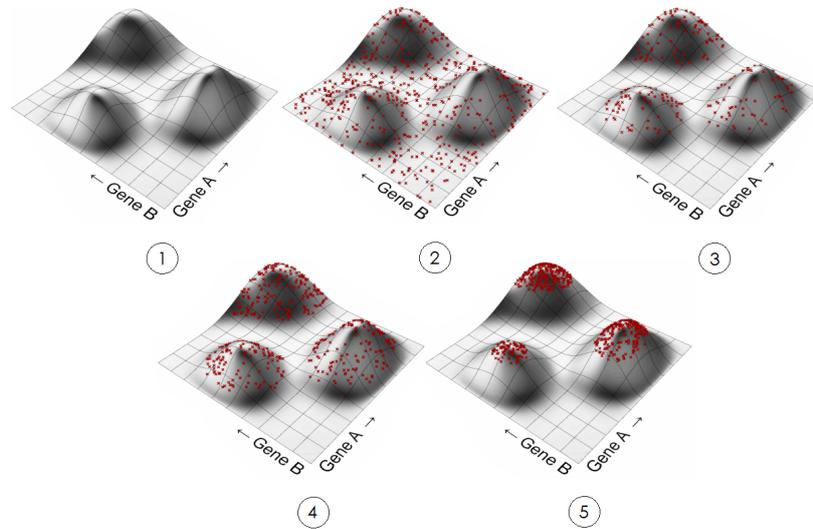


Figure 2.3: Fitness landscape in evolutionary optimisation process (Rutten, 2010)

As a population-based global approach, genetic algorithms are well suited for solving multi-objective optimisation problems (Konak et al., 2006). Genetic algorithms are able to simultaneously search different regions of a solution space, which enables them to find diverse Pareto solutions in non-convex, discontinuous and multi-modal solution spaces. On top of that, most multi-objective genetic algorithms do not need the user to scale or weigh objectives. Consequently, genetic algorithms are the most popular heuristic approach for multi-objective design and optimisation problems (Konak et al., 2006).

Within these multi-objective genetic algorithms there are many different algorithms, which mostly differ based on their fitness assignment procedure, elitism (best found solutions so far survive to the next generation) and/or diversification approaches (Konak et al., 2006). These aspects are all important to approximate the Pareto front. For almost each of these characteristics, a consideration has to be made between the computationally efficiency and accuracy of the algorithm. A. Konak (2006) shows a table of different well-known algorithms with their advantages and disadvantages, evaluated on these points. All comparative studies on multi-objective genetic algorithms however agree, that elitism and diversity preservation mechanisms improve performance of the algorithm. They do however come at a computational cost (Konak et al., 2006).

2.4 Multidisciplinary design optimisation in the AEC industry

Multidisciplinary design optimisation techniques are applied in several fields, such as aerospace and automotive engineering. Nonetheless, their application in the architecture, engineering and construction industry has been modest up to now (Flager et al., 2009). Earlier research however (Flager and Haymaker, 2007; Geyer, 2009; Gerber and Lin, 2014; Yang et al., 2015; Jansen et al., 2014) showed MDO approaches could become beneficial in the AEC industry as well.

2. Multidisciplinary design optimisation

In Chapter 1, a description was given of an “average” design process in the AEC industry. Several difficulties were addressed, which resulted in an insufficient number of alternatives considered during the conceptual design phase. Although Building Information Modelling and parametric design already shown potential and improvements in this design process, application in other fields indicate that multidisciplinary design optimisation could be a next step towards better informed design decisions.

Flager and Haymaker (2007) show an example about a Boeing (multinational corporation in the aerospace branch) project, for which a hypersonic vehicle had to be designed. Preliminary design was critical for this project, as close integration between different disciplines was required to achieve the desired performance level. After six years of project work using sequential design methods (similar to those used in the AEC industry), the design team was unable to come up with a design which met the requirements. In 2002, Boeing began to use a set of technologies and methodologies to support multidisciplinary analysis, such as parametric geometry definition, integrated design schemes, automated discipline analysis and multidisciplinary optimisation. This would lead to an improved process and product performance. The MDO process produced no less than 98 different vehicle configurations and analysed 3.900 engine inlet flow paths in a fully automated loop over the course of six days. As comparison; in the previous eight years of the project, only 12 vehicle configurations and 116 engine inlet flow paths were analysed to reach a baseline design. The MDO process improved the baseline design by 39%.

This example shows that multidisciplinary design optimisation can be a powerful tool to support the design process, also in the AEC industry. Especially for complex design problems as in the example, MDO can help finding better performing and more integrated solutions (Flager and Haymaker, 2007). However, there are significant differences between the AEC and aerospace industry, making MDO (more) difficult to apply on building design problems. The concept of overall multidisciplinary performance for a building is ill-defined, since the performances are not all quantifiable (e. g. aesthetics). On top of that are (most) buildings one-of-a-kind designs. This means that they have a unique problem statement, which makes the standardisation of a solution strategy difficult. For different designs, different disciplines are decisive and the interactions between different disciplines are different per building, much less ‘organised’ than in designs in other engineering fields (Jansen et al., 2014). Finally, building design problems usually require many parameters to evaluate the performances of the different disciplines, which complicates automated optimisation in an acceptable amount of time (Jansen et al., 2014).

These differences are to be kept in mind, as they might influence the applicability of a multidisciplinary design optimisation for building design. Although design optimisation cannot replace the entire design and calculation procedure, it is however capable of supporting the designer in finding a good performing solution (Geyer and Rueckert, 2005). To summarise, a multidisciplinary design optimisation approach can lead to the following improvements in the design process (Bradner and Davis, 2013):

- Faster, easier and more elaborate exploration of the design space.
- Increased integration between different design and engineering disciplines.
- Better understanding of the trade-offs between different (conflicting) objectives.
- Better understanding of the cause and effect of design decisions.

2. Multidisciplinary design optimisation

These improvements result in better informed decision making, which ultimately could lead towards better performing designs. However, an appropriate approach is needed to allow architects and engineers to benefit from multidisciplinary design optimisation. The approach should therefore satisfy the following requirements:

- The approach should have an acceptable computation time. Ideally, the Pareto front of a problem would be found within (roughly) 12 hours. In that way, the optimisation could be started at the end of a working day and is finished at the start of the next day. As a result, users do not have to wait for the optimisation process to be finished. The number of solutions which need to be evaluated before the Pareto front is estimated accurately, depends on the size of the design space. If it would need 10.000 solutions, the computation time for one solution should be around 4 seconds. It should be noted, that the 'computation time' for a single solution in traditional design approaches is over one month (Flager and Haymaker, 2007).
- The approach should be usable for architects and engineers, not just for researchers and developers. This means that architects and engineers should be able to use this approach without further assistance or research. Ideally, the approach consists of applications with which architects and engineers are already familiar.
- The results of the optimisation process (designs and performances) should be clearly visualised. The visualisations should thus entail sufficient information for users to evaluate, understand and compare different solutions themselves.
- The approach should be applicable for different disciplines and design problems. Consequently, the approach should not be exclusively focused on the case study which it is tested with.

Applications of different approaches in the AEC industry can be found in some earlier studies. Some examples will be shortly illustrated and compared based on the stated requirements.

F. Flager and others show a case study in which both the structure and the energy consumption was optimised. A Process Integration and Design Optimisation (PIDO) comprise software was used to automate and manage the setup of simulation and analysis, to integrate and coordinate the analyses and to optimise several design aspects by iterating analyses towards a specified set of target conditions. Commercially available ModelCenter was used to allow the user to bring commercial or proprietary software tools in a common environment (Flager et al., 2009). Outcome after 34 hours were 5600 design alternatives, which means about 22 seconds were needed to evaluate one alternative. The implementation of the workflow was however difficult, software development expertise was needed to implement the different analysis tools, which is not a very common expertise for architects and engineers. Other advanced tools were used for visualisations. This means, however applicable for different disciplines and designs, the implementation of it costs a significant amount of time. The authors concluded that PIDO has potential in the AEC industry, however much work is needed to examine the applicability on complex and large scale AEC projects and on determining a way of integrating this approach with conventional tools and methods (Flager et al., 2009).

D. Yang and others propose an integrated process with Rhino/Grasshopper and modeFRONTIER. Grasshopper is a visual programming language integrated with Rhino 3D modelling tools.

2. Multidisciplinary design optimisation

It is known for handling complex geometries and does not require knowledge of programming or scripting (Yang et al., 2015). Furthermore, architects and engineers are already becoming more and more familiar with Rhino/Grasshopper. ModeFRONTIER is an integration platform for multi-objective and multidisciplinary design optimisation, used in many engineering industries. Grasshopper plug-ins were used for the different analyses, focused on the structural, energy and daylight simulations. ModeFRONTIER drives the optimisation process by altering the input values and analysing the simulation outputs. A process resulted in 245 solutions, which each took 3 minutes on average. A key advantage of modeFRONTIER is its data analysis environment, which allows statistical assessment of the data and visualises the results in a meaningful manner (Yang et al., 2015).

D. Jansen and others present a developed multidisciplinary optimisation system for the AEC industry. It consists of two main components, an MDO framework and a distributed cloud-based analysis framework (Jansen et al., 2014). The use of this cloud-based framework means the system is capable of rapidly evaluating the different design alternatives. The modelling layer of the framework is built as a plug-in for the earlier mentioned Grasshopper application for Rhinoceros. It allows the user to model the system, interpret the disciplinary models and define performance equations in a relatively familiar environment. On top of that, non-quantifiable aspects (e. g. aesthetics) can be judged by the designer through the geometrical representation. The optimisation layer allows the user to influence the optimisation process by instantiating new scenarios at will to effectively explore different areas of the design space for different scenarios (Jansen et al., 2014). In each scenario several single objective optimisation processes (custom genetic algorithm) are treated in parallel where coupling between different objectives is organised by the user. This approach allows the designer to explore both the performance of different single disciplines and the trade-off between them (Jansen et al., 2014). The generation layer will evaluate the different design alternatives produced by the optimisation layer, and is based on a scalable distributed computing model. Finally, the visualisation layer shows the processed information to the user in parallel plots and an objective comparison graph. The current implementation of the system does however not contain as much functionality (yet) as was envisioned. For now the modelling layer is limited to only two specific disciplinary simulations and the visualisations and post processing are rather static and the use of Rhinoceros as generator of geometrical representations is not exploited to its full potential (Jansen et al., 2014).

M. Neill and others propose a prototypical tool, which is capable to perform trade-off studies within a parametric modelling environment (Neill et al., 2009). The goal of the proposed design tool is to improve upon more traditional methods which use external programs outside the CAD model to perform optimisation routines. This improvement was accomplished by fully integrating the optimisation methods into pre-existing industry software, consequently bypassing the learning curve needed for using a new program and the lag time between updating different models (Neill et al., 2009). Key component to the chosen work flow, is the human-computer feedback loop (Neill et al., 2009). The human component consists of entering the design variables, objective functions, constraints and preference functions. The computer then searches for the Pareto optimal set, after which the designer is able to modify the earlier mentioned input, as a reaction to the found solutions. Consequently, the tool gives a certain insight and control to the designer (Neill et al., 2009).

T. Uijtenhaak and J. L. Coenders present the opportunities that a dashboard-based system

2. Multidisciplinary design optimisation

could provide to gather information on design alternatives and display this information in a way it will help designers to make decisions by the use of multidisciplinary design optimisation (Uijtenhaak and Coenders, 2012). With the help of this system, the user is able to understand the influence of various involved disciplines. On top of that the tool could help explaining other parties why choices are made and how they affect the design. The presented tool consist of three separate elements (Uijtenhaak and Coenders, 2012). Firstly, alternatives are generated with the help of parametric software (such as Grasshopper). The different alternatives, satisfying the geometric constraints, are analysed and assessed for the various disciplines. Next, a database with all this data will be used to feed this to the dashboard to be presented. Selections can be made via search criteria and constraints, to filter obsolete solutions. Finally, the user is able to analyse and interpret the results with the help of the dashboard (Uijtenhaak and Coenders, 2012).

Octopus is another plug-in for the Grasshopper application for Rhino and applies evolutionary principles to parametric designs made in Grasshopper. It allows the search for multiple objectives at once, producing a set of optimised trade-off solutions between the extremes of each objective (Vierlinger, 2012). It was part of a range of tools developed at the University of Applied Arts Vienna and Bollinger+Grohmann Engineers. It makes use of multi-objective evolutionary algorithms and it's user interface is based on the user interface of Galapagos, the standard optimiser in Grasshopper. This means that it is relatively easy for users who are familiar with Grasshopper. It allows the user to change objectives during the search, choose preferred solutions and to save solution states (Vierlinger, 2012). A major advantage of working within Grasshopper, is the representation of the design in Rhinoceros. In that way the aesthetics of a design can be evaluated as well, which is very important in the AEC industry. On top of that it is free of charge, so usable by everybody. As Octopus only evaluates different objectives, other Grasshopper plug-ins are needed for the analyses of the different disciplines. Drawback of this 'simple' approach is the computation time, which becomes a problem when designs and analyses are comprehensive.

As stated earlier, it was chosen to research an approach with 'Octopus', as it is easy to use and applicable for various disciplines. This means such an approach could be valuable for many different design problems faced in the AEC industry and is in addition also usable for architects and engineers themselves. The drawback of the long computation time will be evaluated together with its applicability for large complex design problems.

3

Acoustic and structural analysis

In this chapter, the different analysis methods that are needed for stadium case study are explained, illustrated and tested. As stated, the case study will focus on the acoustic and structural performance of the roof of the Donbass Arena. Therefore, this chapter is divided in two main sections; the acoustic analysis and the structural analysis.

3.1 Acoustic analysis

Acoustics are becoming increasingly important in stadiums as stadiums are nowadays often multi-purpose, which means they are designed to be easily usable for multiple types of events. Due to their size and complexity, the investment for such buildings are significant. By being able to host as many (different) events as possible, a stadium could possibly justify these costs. Consequently, stadiums are often not only used for sporting events but for concerts as well. However, the acoustic quality of stadiums during concerts (and sporting events) can, in some cases, be experienced as rather poor (Vennard, 2013).

The acoustic design of fully and extensively roofed stadiums starts with their shape and materials, as the effects of sound reflections and noise build-up can be significant (John et al., 2013). These aspects of a building are essential to the performances of other disciplines as well, which means an investigation of the trade-offs between these disciplines could be valuable when making design decisions. Computational acoustic simulation software is capable of estimating the acoustic response with a good degree of accuracy, but these tools on their own will not help designers to explore the different building shapes (Echenagucia et al., 2014). In combination with parametric and associative design, these tools are however capable of evaluating the acoustic “quality” of different designs and could help to find optimal design(s).

3. Acoustic and structural analysis

In this section, the approach for the acoustic analysis methods will be illustrated. First the analysis methodology will be explained, after which some tests are conducted to evaluate the chosen approach.

3.1.1 Ray tracing

In physics, ray tracing is a method for calculating the path of waves or particles through a system which can contain regions with different propagation speed, absorption characteristics and reflecting surfaces. Consequently, wave fronts can bend, change direction and reflect off surfaces. Ray tracing idealises the wave front by a (large) discrete amount of lines, so called 'rays' (Wikipedia, 2015b). Rays are assumed to propagate from a source in straight lines, until they encounter a change in medium or reflecting/absorbing surface. At these points, the new direction of the ray is calculated after which the process repeats itself. Goal of the approach is to find the paths from the sound source to the receiver(s) (Pelzer et al., 2014). The amount of rays is dependent on the number of rays that are needed to understand the behaviour of the system (Wikipedia, 2015b). The amount of rays has a large influence on the accuracy of the analysis.

The algorithms of most typical acoustical programmes are based on this principle (Vorlander, 2010). This means that some sound properties, such as diffraction and interference are neglected. Description of the sound field is thus reduced to energy, transition time and direction of rays. Such an approach is solely representative when the dimensions of the room are large compared to the wavelengths (Vorlander, 2010). As long as frequencies are not very low, wavelengths are negligibly small compared to stadium dimensions.

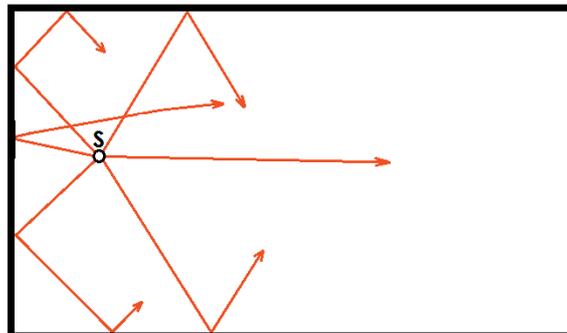


Figure 3.1: Ray tracing concept with 'S' as sound source

Every time a 'ray' encounters a surface, part of its energy (E) is absorbed by or transmitted through the surface (α) and the rest of its energy is reflected back into the room. The amount of sound which is absorbed is dependent on the absorption coefficient of the material of the reflecting surface.

The direction of the reflected part of the ray is dependent on the incoming angle of the ray, the shape of the reflecting surface and the roughness of the surface. Specular reflections ($(1 - \alpha)(1 - s)$) is the main reflection type and can be compared to light which is reflected in a mirror. The angle of incidence (θ) is equal to the angle of reflection. However, surfaces can

be irregular of texture, which leads to scattering $((1 - \alpha)s)$ of the sound. Figure 3.2 shows the reflection.

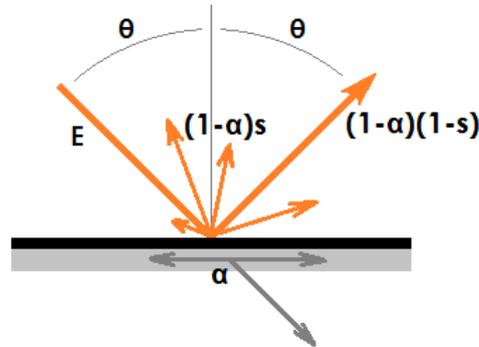


Figure 3.2: Reflection (specular and scattered) and absorption of rays

Geometric acoustic approaches, such as ray tracing, provide approximations of the acoustical environment in a room. Pelzer et al. (2014) state that geometric approaches are more suitable for optimisation than approaches applying wave acoustic theory, because of their significant shorter computation time. Within geometrical methods, two main methods can be distinguished: earlier described ray tracing and image source modelling. Image source models construct reflection paths by mirroring the source in the planes of the reflecting surfaces, to create mirror sources. The total sound field contains the contributions of all valid mirror sources. However, the number of image sources quickly expands when the reflection order is increasing and when geometry involves complex shapes. Considering the curved geometry of the stadium, this research employs a ray tracing approach for the acoustical analyses.

3.1.2 Sound pressure level, sound power level and sound intensity

The acoustic value that will be analysed throughout the building is the sound pressure level. Sound pressure is the difference between the ambient atmospheric pressure and the pressure caused by a sound wave (Gracey and Associates, 2015). The SI unit of sound pressure is pascal (Pa). The sound pressure level (SPL) is a logarithmic measure of the effective pressure of a sound, relative to a reference value (Gracey and Associates, 2015). This reference sound pressure is also known as the threshold of human hearing. It is measured in decibels (dB) and is denoted as L_p .

$$L_p = 20 \cdot \log \left(\frac{p}{p_0} \right)$$

$$\begin{aligned} \text{with : } \quad p &= \text{sound pressure} \\ p_0 &= \text{reference sound pressure} \\ &= 20 \cdot 10^{-6} \text{ Pa} \end{aligned} \tag{3.1}$$

3. Acoustic and structural analysis

To be able to calculate the sound pressure levels throughout the building with the help of ray tracing, first some other acoustic properties need to be explained. Sound power (P) is the rate at which sound energy is emitted per unit of time (Gracey and Associates, 2015) and has the SI unit watt (W). Unlike sound pressure, sound power is neither dependent on the room or distance.

$$P = P_0 \cdot 10^{0,1L_w} \quad (3.2)$$

$$\text{with : } P_0 = 10^{-12} W$$

The sound power level (SWL) is the logarithmic measure of sound power, is signified with L_w and measured in decibels (dB) (Gracey and Associates, 2015). Note that this formula is only applicable for point sources.

$$L_w = L_p + |10 \log \left(\frac{Q}{4\pi r^2} \right)| \quad (3.3)$$

$$\begin{aligned} \text{with : } Q &= \text{directivity factor} \\ & (= 1 \text{ for omnidirectional}) \\ r &= \text{distance from source} \end{aligned}$$

When ray tracing is applied, the sound power of the force is divided by the number of rays used in the calculation. Each time a ray reflects of a surface and when it moves through air, it loses energy. The remaining sound power (P_i) in a ray can be determined as is shown in Equation 3.4 (Elorza, 2005).

$$P_i = \frac{P_s Q}{N} e^{-mL} \prod^J (1 - \alpha_j)$$

$$\begin{aligned} \text{with : } P_s &= \text{source sound power} \\ Q &= \text{directivity factor} \\ & (= 1 \text{ for omnidirectional}) \\ N &= \text{total number of rays} \\ m &= \text{absorption coefficient} (= 0.012 \text{ dB/m}) \\ L &= \text{travel distance of } i^{\text{th}} \text{ ray (m)} \\ J &= \text{number of reflections during ray path} \\ \alpha_j &= \text{absorption coefficient at } j^{\text{th}} \text{ reflection} \end{aligned} \quad (3.4)$$

Sound intensity (I) is defined as sound power per unit area, so the SI unit is (W/m^2) (Gracey and Associates, 2015). Sound intensity can be converted to the sound pressure level.

$$L_p \approx 10 \log \left(\frac{I}{I_0} \right) \quad (3.5)$$

with : $I_0 = \text{reference sound intensity}$
(= $10^{-12} W/m^2$)

In the following paragraphs the implementation of these formulas in a ray tracing tool is described. This tool is used to estimate the sound pressure levels throughout the stadium bowl.

3.1.3 Custom ray tracing tool and benchmarking

As stated earlier, Rhino and its plug-in Grasshopper were chosen to model the building and to perform the multidisciplinary design optimisation in. Within Grasshopper a ray tracing method is used to analyse the acoustics and in more detail the sound pressure levels. The pressure levels are determined with the help of the aforementioned formulas. The procedure of calculation is shown below.

1. Choose the starting sound power level which is emitted by the source(s).
2. Divide the source sound power by the number of rays used in the model.
3. Calculate the sound paths of the rays. See the pseudo code in Appendix A for the exact calculation method. Note that simplifications are made in the ray tracing process as the needed computation time should remain within limits. Scattering is not taken into account, to make calculation of the reflection angles more simple.
4. Calculate remaining power in each ray after losses by absorption by the reflecting surfaces and air.
5. The remaining power of all rays which intersect with a receiving surface are summed and divided by the area of the receiver to determine the sound intensity at each receiver. Note that sound intensity is defined as the sound energy passing per second through a unit area held perpendicular to the direction of the propagating sound waves (Gracey and Associates, 2015). This is however simplified in this calculation to spare computation time.
6. The sound pressure level is finally calculated for each receiver surface.

3. Acoustic and structural analysis

Several ray tracer tools are available to use in either Rhino or Grasshopper. Most of them however appeared either too comprehensive or not functioning correctly when a large number of rays (over 10.000) were used. With the size of the stadium in mind, such an amount of rays is definitely necessary to be able to estimate the sound paths. That is why it was chosen to make a custom Python script to model the rays within Grasshopper. The script of this custom tool, together with the SPL-Estimator Grasshopper component, is shown in Appendix A.

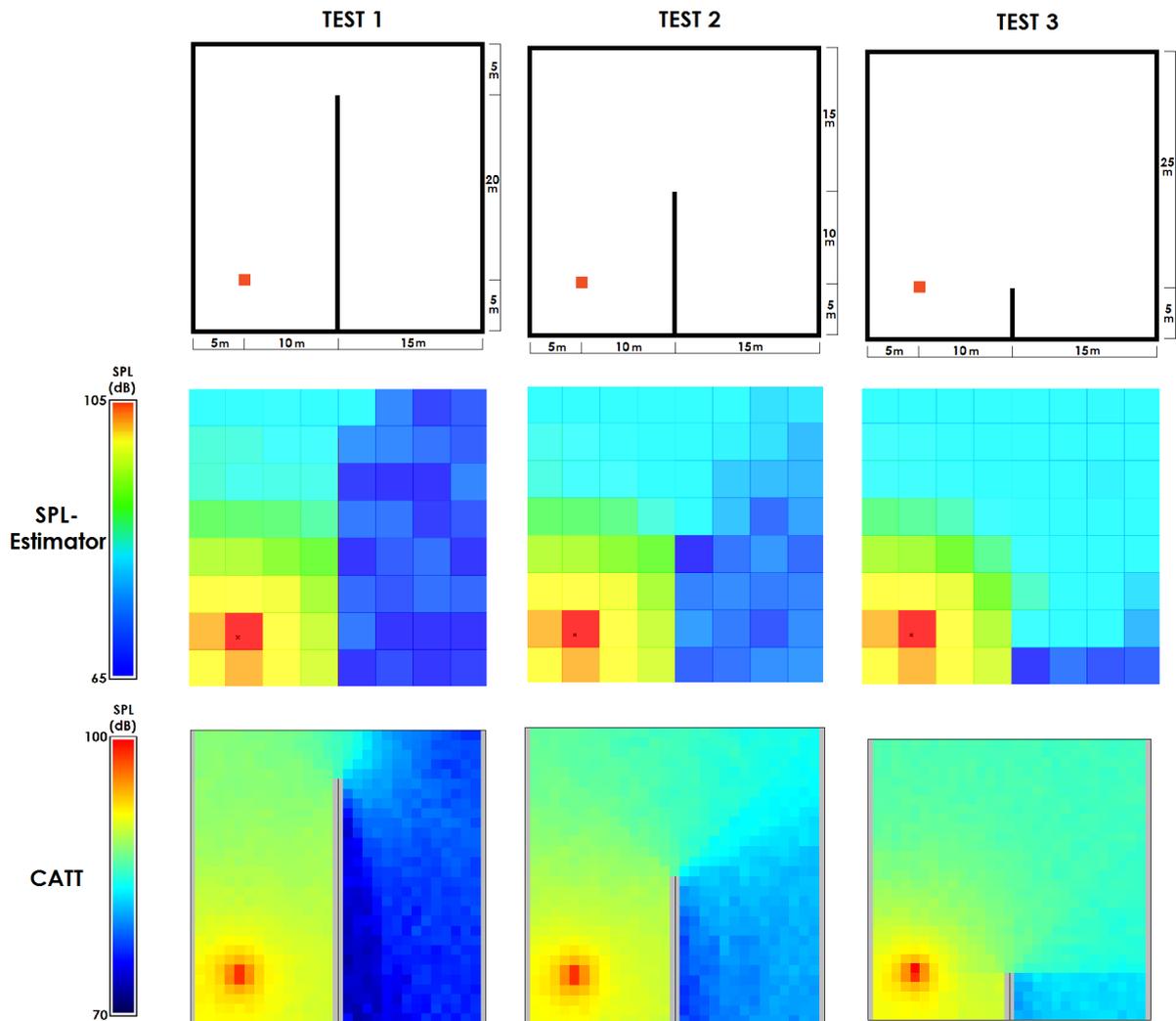


Figure 3.3: Test results with room shapes (first row), SPL-Estimator results (second row) and CATT results (third row)

To test whether or not SPL-Estimator is accurate, a benchmark case has been set up to compare the results found by the tool with an existing commonly used acoustic analysis programme, CATT Acoustic. CATT acoustic is software for room acoustics prediction (CATT, 2015). Figures 3.3, 3.4, 3.5 and 3.6 show the room shapes and results of the three benchmark cases. Note that the amount of divisions in the receiving surface is limited for SPL-Estimator, as a higher number of surfaces led to problems with computation (for this amount of rays).

3. Acoustic and structural analysis

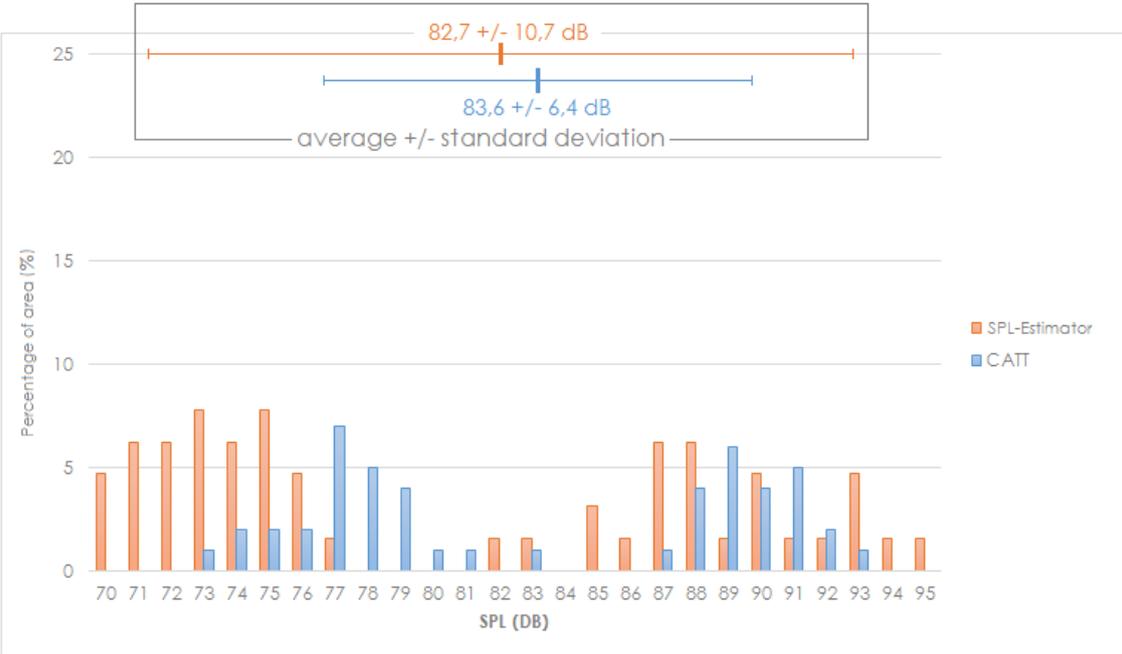


Figure 3.4: Histogram of sound pressure levels test case 1 with average and standard deviation

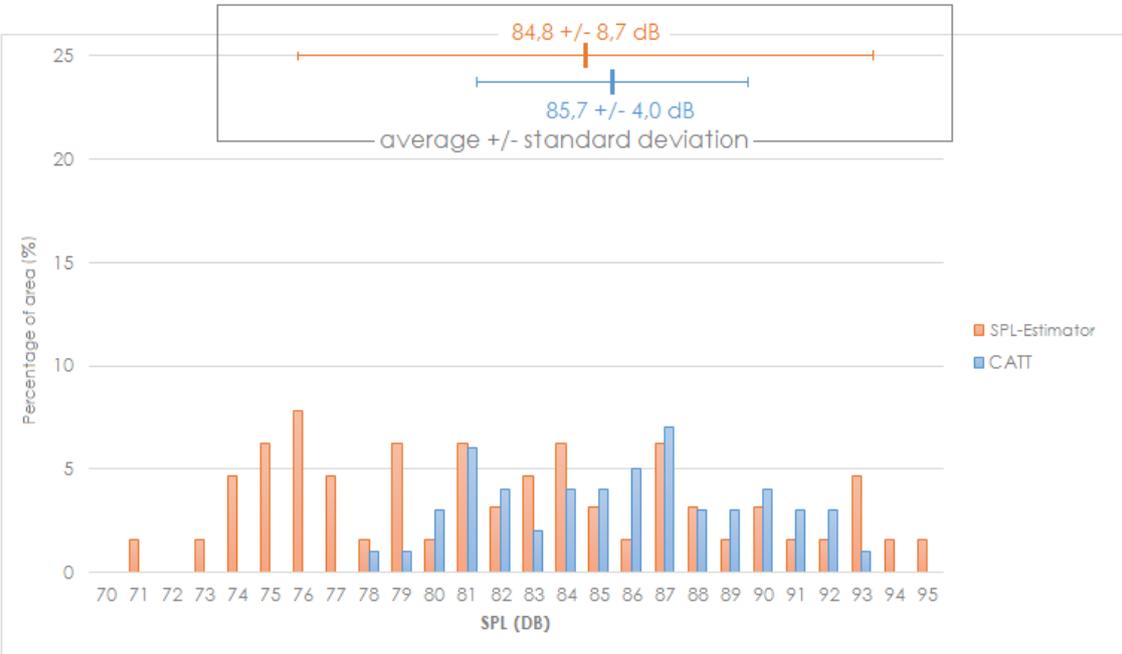


Figure 3.5: Histogram of sound pressure levels test case 2 with average and standard deviation

3. Acoustic and structural analysis

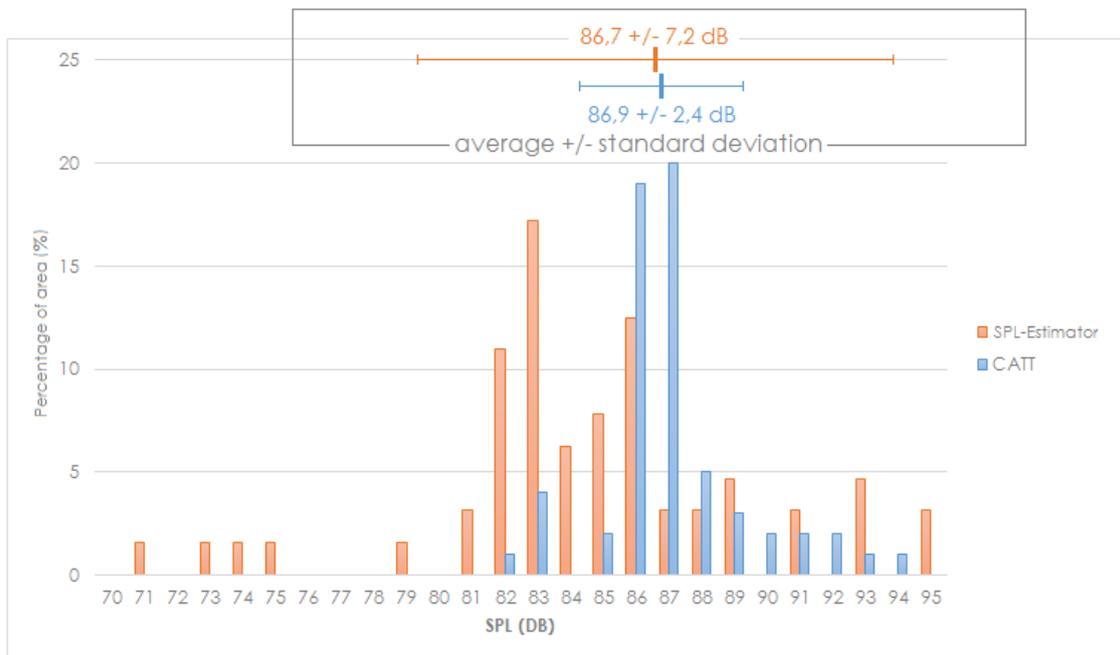


Figure 3.6: Histogram of sound pressure levels test case 3 with average and standard deviation

Some general conclusions, which apply on all test cases, can be drawn from these results. These are enumerated below.

- The average sound pressure levels are estimated lower by SPL-Estimator than with the CATT Acoustic analyses.
- The standard deviation is overestimated by the SPL-Estimator in comparison with CATT. This is also the reason why a bigger range was chosen for the custom tool in Figure 3.3.

These differences can be explained by several reasons, however further examination of the influence of some important tool parameters was conducted to get more insight. In Figures 3.7 and 3.8 the influences of the number of receiving surfaces and rays on the average value and standard deviation of the sound pressure levels are studied.

Figure 3.8 shows that, although very important for the precision level of the analyses, the number of receiving surfaces does not have a clear influence on the average and standard deviation of the sound pressure levels. In contrast, the number of rays has a very clear effect on these values. Figure 3.7 shows that beyond a particular amount of rays, the average and standard deviation are becoming close to constant. However, below this value the values vary considerably. This means that an analysis with an insufficient number of rays is not representable. Note that this transition value is different for every room shape.

3. Acoustic and structural analysis

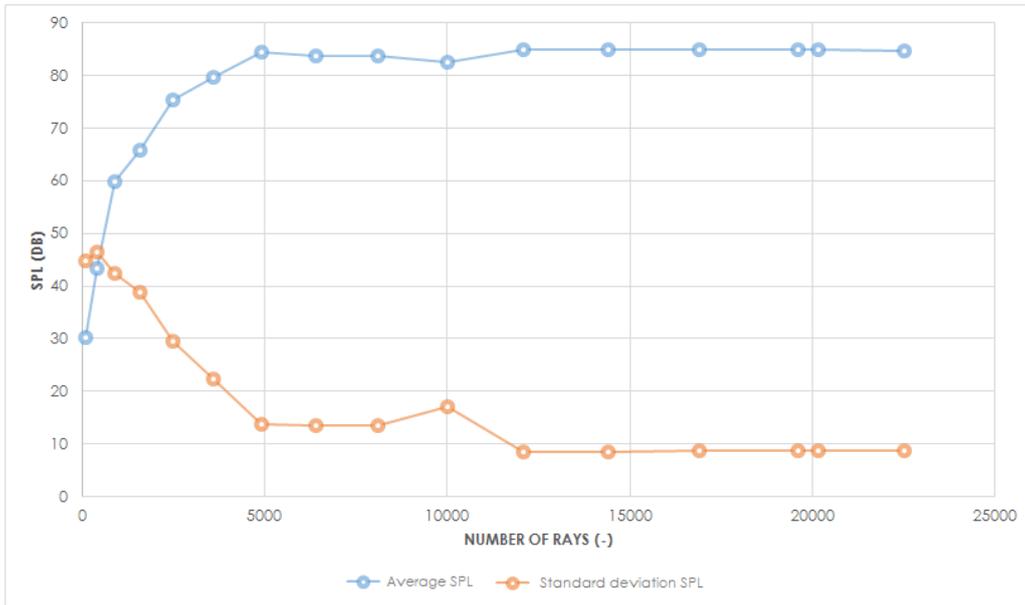


Figure 3.7: Influence of the number of rays on SPL values (number of receiving surfaces is constant at 64)

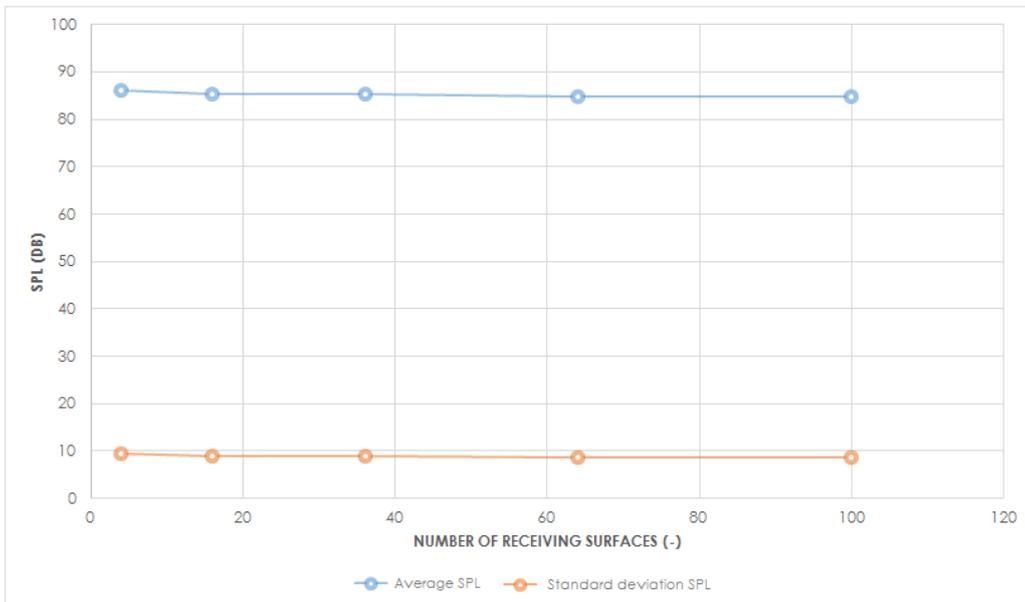


Figure 3.8: Influence of the number of receiving surfaces on the SPL values (number of rays is constant at 20,164)

3. Acoustic and structural analysis

This means the deviations between the results of SPL-Estimator and the results of CATT Acoustic can be caused by the following differences.

- CATT Acoustic uses a more extensive and precise (hybrid) calculation method than SPL-Estimator.
- The maximum number of bounces is insufficient.
- Start vectors of the source are not perfectly divided over all directions in SPL-Estimator.
- Simplifications made for SPL-Estimator lead to inaccurate results.
 - Some rays escape from the room at the points where two reflecting surfaces meet.
 - Rays do not bounce from receiving “target” surfaces.
- Geometry in CATT Acoustic needs to be defined in meshes, while the geometry has to be defined in NURBS surfaces for SPL-Estimator. This is a more accurate approximation for the geometry (Echenagucia et al., 2013). Although the meshes for the test geometries are equal to the NURBS surfaces, this difference in approach can lead towards deviations between the results in other cases.

Consequently, improving the accuracy of SPL-Estimator will lead to more computation time. However, computation time could most probably be the main drawback of a multidisciplinary design optimisation approach. Although the calculated sound pressure levels can be deviating from the actual levels, the estimations are sufficiently representable for the optimisation process when their limitations are kept in mind and the results are validated. The trend between the design and its overall acoustic performance compared with other designs is quantifiable by SPL-Estimator. However, when exact estimations of the sound pressure at certain points in a room are needed, other existing tools are more appropriate.

In Section 5.1, the acoustic objective and acoustic design variables for the case study are further explained.

3.2 Structural analysis

The scale of a stadium alone ensures that its structure is an essential part of its appearance and design. Also the amount of material which is needed for the structure is enormous and consequently, optimising the structure can lead to a significant amount of material savings. As the structure has such an impact on the appearance, it is important that it is integrated in the architectural concept of the building and thus has, together with other disciplines, a large influence on the overall shape of the building. On top of that, the shape has major influence on the way a structure 'works' and performs. Structural optimisation is able to examine the influence of the geometry on the structural performance of a building.

A structure, in mechanics, can be defined as "any assemblage of materials which is intended to sustain loads." Structural optimisation is thus the search for the best way of assembling materials to sustain the loads (Christensen and Klarbring, 2009). The optimisation problem can be divided in three classes (Figure 3.9), although it is common that an optimisation consists parts of several of these classes.

- Sizing (or section) optimisation; which searches for the optimal cross-sections or thicknesses for a design.
- Shape optimisation; which searches for the optimal geometry of a structure. Note that the connectivity between different members does not change.
- Topology optimisation; which searches for the optimal structural topology within a given region. This is the most general form of structural optimisation.

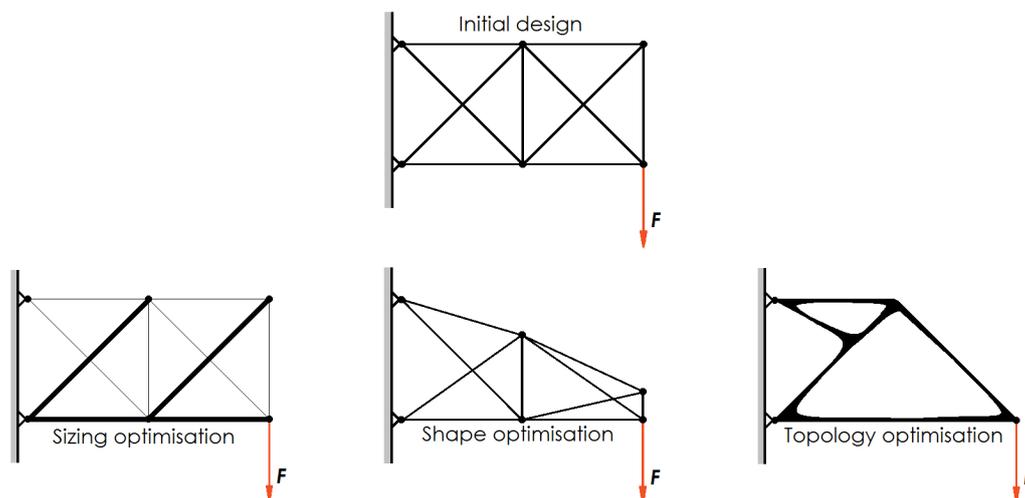


Figure 3.9: Structural optimisation classes

The optimisation process in this research uses a combination of two of these approaches, namely sizing and shape optimisation. This will be illustrated in this section. The structural analysis methodology will be explained and evaluated.

3. Acoustic and structural analysis

3.2.1 Karamba

Karamba is a finite element program to predict the behaviour of structures under external loading. It has been developed by Clemens Preisinger in collaboration with Bolling + Grohmann Engineers. Karamba is fully embedded in the parametric environment of Grasshopper (Preisinger, 2013, 2015), which makes it very easy to combine it with the parametric modelling and optimisation algorithms within Grasshopper and thus leads to a smooth flow of data between the structural and geometric models. On top of that, parallelisation of some calculation steps and calculating results on demand lead to a reduction of computation time in comparison with other finite element programs (Preisinger, 2013). This is very important when a large number of designs has to be evaluated during an optimisation process. It has been developed for the use within a parametric architectural design environment, thus not solely for an engineering setting (Preisinger, 2013), and consequently Karamba is a suitable program for this research.

Creating a basic static beam model in Karamba contains six steps (Preisinger, 2015). These steps are used for setting up the structural model in these research. The first step is to create a geometry in Rhino/Grasshopper, consisting out of points or lines (1). This geometry resembles the structure of the building design. The geometry is then converted by Karamba into beam members (2). These members can be given different cross-sections and material properties. The next step is to define the supports and loading (3). Besides gravity loading, also point and distributed loads can be applied in different directions and with a different magnitude. Also different load combinations can be applied. Supports restrain translations and rotations in different global directions at prescribed nodes. Next, Karamba assembles the elements, their properties, the supports and loadings in a model (4). The structural model can then be analysed (5) by different algorithms. The standard analysis component calculates the response of the model under the external loads. Besides that, other numerical evaluation options are calculations for second order theory, eigen modes, natural vibration modes, large deflections and different iterative optimisations (Preisinger, 2015). The analyses which are used for this research are elaborated further in Section 3.2.2. Finally, the results such as stresses, utilisation of elements and deflections can be viewed (6) with different visualisation options and used for post processing. It can be chosen to evaluate different load combinations separately or together. Which results will be used in the optimisation process is explained in Section 5.1.2. Figure 3.10 shows an example of a structural model in Karamba.

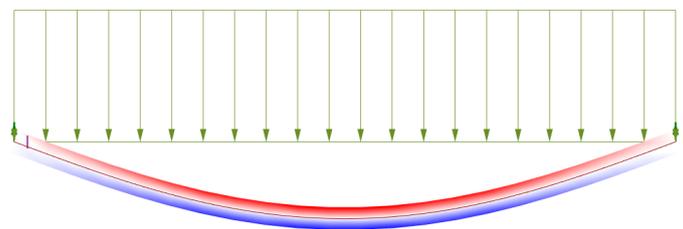


Figure 3.10: Example of a structural model in Karamba, with the loading and utilisation (red shows compression, blue shows tension) of the beam

3.2.2 Analyses and algorithms

Several types of analyses and algorithms which can be used in Karamba are mentioned in the section before, but not all of them are used in this research. The analyses and algorithms which are used for the structural analysis and optimisation of the case study are elaborated in this section.

The main analysis component which is used, is the so called “Th1” component, which computes the deflections, total gravity force and internal deformation energy of the model for the different defined load combinations (Preisinger, 2015). These are numerical outcomes and all possibilities to assess the performance of a structural design. The outcome of the analysis can however also be shown in the Rhinoceros model. The component “ModelView” is able to show the deflections of the model, supports and loadings. The component “BeamView” shows internal force diagrams, utilisation of elements and axial stresses in the elements. The utilisation of the elements can be broken down to the internal forces as well, by using the “Utilisation” component. This component determines the internal forces according to the EN1993-1-1 (European Committee for Standardization, 2005). The exact calculation method cannot be viewed, with the exception of some intermediate values. The component evaluates the normal forces (N), shear forces (V_y , V_z), torsional moments (M_t) and bending moments (M_y , M_z). Note that buckling is taken into account (Preisinger, 2015). The component is able to return the maximum utilisation factor for each member.

In the optimisation process, the “Optimise Cross Section” component is used. After some tests were conducted on different workflows (see Section 4.2), it turned out that the optimisation process was able to find more feasible design alternatives when the cross-sections were modelled as state variables (Section 2.2) instead of global variables. This means the cross-sections are not controlled by the global optimisation algorithm, but by a separate structural optimiser.

The “Optimise Cross Section” component defines the cross-section of each element in such a way that their load-bearing capacity is sufficient for all load combinations (Preisinger, 2015), constraint by the (given) maximum utilisation and displacement of the structure. The component does this by first computing the section forces at the prescribed ‘N’ points along all beams, using an initial prescribed cross-section. Next, the first sufficient entry from the given list of possible cross-sections is chosen for each element or group of elements (which all get the same cross-section). The algorithm stops if the second step does not lead to any changes or when the maximum number of iterations is reached. Otherwise the steps are repeated. This iterative process is needed for statically indeterminate structures, at which the section forces depend on the stiffness of the different members. Note that it is important that the given list of possible cross-sections, runs from best to worst possible cross-section, as the component starts at the first entry of the list and proceeds until the first sufficient cross-section is found. Note that it is not possible to iterate through lists with different cross-section families. It is for example not possible to use a list which contains both H-profiles and circular hollow sections.

3.2.3 Structural benchmarking

Before applying the Karamba plug-in as structural analysis and optimisation for a large and complex structure, it is important to test the programme on simpler cases to investigate its

3. Acoustic and structural analysis

potential inadequacies or limitations. The results found with Karamba will be compared to the results found with the structural analysis software GSA (Oasys Limited, 2016).

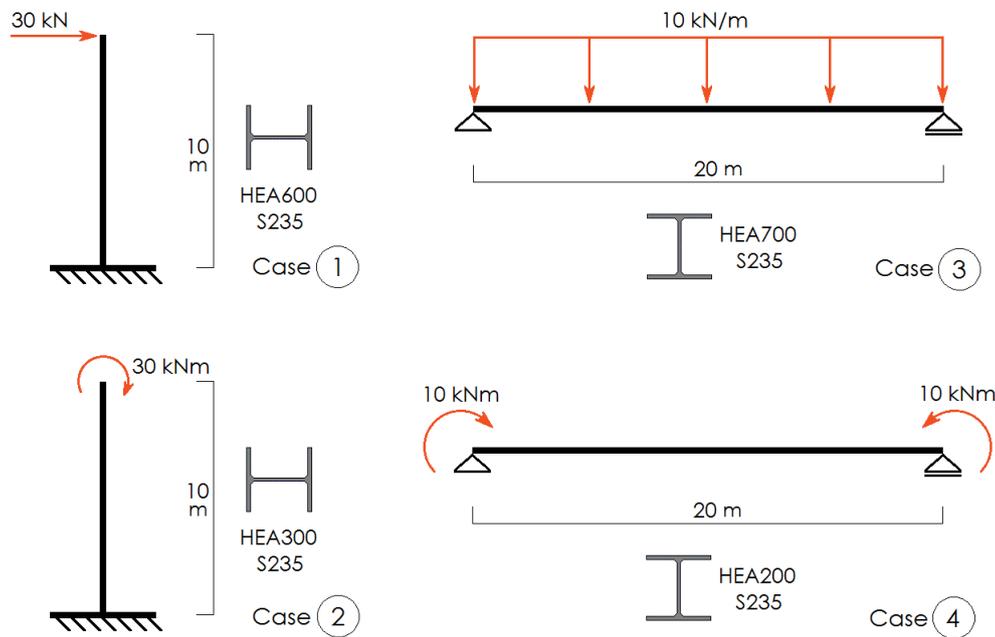


Figure 3.11: Simple structural benchmarking checks

The first four basic analysed cases are shown in Figure 3.11. The comparison of the results are shown in Table 3.1. The results that are compared are the maximum bending moment and bending moment line (to verify the set-up of the models), maximum deflection, maximum stresses and maximum utilisation.

	Case 1		Case 2		Case 3		Case 4	
	Karamba	GSA	Karamba	GSA	Karamba	GSA	Karamba	GSA
M_{max} (kNm)	300	300	30	30	500	500	10	10
u_{max} (mm)	34	35	39	40	47	48	64	66
σ_{max} (N/mm ²)	62.7	62.7	23.8	23.8	80.0	80.1	25.7	25.7
Max. util (-)	0.66	0.27	0.21	0.11	0.99	0.90	0.29	0.29
Difference	39 %		10 %		9 %		0 %	

Table 3.1: Comparison outcome of simple benchmark cases

Although most results calculated with Karamba differ little from the results found with GSA, the maximum utilisation in first three cases are deviating significantly. Karamba is in all cases more conservative than GSA and an optimisation making use of this utilisation will thus lead to con-

servative and suboptimal designs. Further investigation shows that the deviation in utilisation is caused by differences in the elastic critical moment. This value is needed for calculating the lateral torsional buckling resistance according to Eurocode 3 (European Committee for Standardization, 2005) and for each of these cases, lateral torsional buckling is normative. However, the Eurocodes do not give a clear advice about how to determine this parameter. This explains the differences between the elastic critical moment and thus the maximal utilisation.

LTBeam is used together with hand calculations to determine the actual elastic critical moment for each case and to check whether Karamba or GSA uses the correct value for computing the lateral torsional buckling resistance. LTBeam is a software which specifically deals with the elastic lateral torsional buckling of beams, developed by CTICM (Centre Technique Industriel de la Construction Metallique) in the frame of an European research project (CESDb, 2013). Table 3.2 shows the values of the elastic critical moment for each case, determined by Karamba, GSA and LTBeam. The differences are

	LTBeam	Karamba	Difference	GSA	Difference
Case 1 - M_{cr} (kNm)	1826	464	-75 %	1829	1 %
Case 2 - M_{cr} (kNm)	155	157	1 %	580	274 %
Case 3 - M_{cr} (kNm)	629	553	-12 %	613	-3 %
Case 4 - M_{cr} (kNm)	34	35	3 %	34	0 %

Table 3.2: Comparison elastic critical moment of different simple benchmark cases (difference in comparison with LTBeam result)

The elastic critical moment can be determined with the help of SN003A (Bureau, 2005) and SN006A (Galea, 2005). The first cases can be computed by the following formulas from SN006A, as it is focused on cantilevers.

$$M_{cr} = C \cdot M_{cr,0}$$

$$\text{with : } M_{cr,0} = \frac{\pi}{L} \cdot \sqrt{EI_z GI_t}$$

$$\kappa_{wt} = \frac{1}{L} \cdot \sqrt{\frac{EI_w}{GI_t}}$$

$$\text{and : } \eta = \text{destabilizing factor } (= 0) \tag{3.6}$$

$$E = \text{Young's modulus } (N/mm^2)$$

$$I_z = \text{second moment of area about the weak axis } (mm^4)$$

$$G = \text{shear modulus } (N/mm^2)$$

$$I_t = \text{torsion constant } (mm^4)$$

$$L = \text{beam length } (mm)$$

$$I_w = \text{warping constant } (mm^4)$$

3. Acoustic and structural analysis

κ_{wt} and η are both needed to determine the C -value, which can be found in Tables 3.1 and 3.2 in SN006A (Galea, 2005). The other parameters are depending on the cross-section, steel properties and the structure. The aforementioned tables are the source for the differences between Karamba and GSA. In case 1 Karamba uses the table with a constant moment to determine C , while the bending moment diagram is actual linear (in this case $C = 2.14$) instead of constant ($C = 0.52$). That is why Karamba underestimates the elastic critical moment. For the second case, it is exactly the other way around. While Karamba uses the correct table to determine C ($C = 0.52$), while GSA uses the table for a linear moment diagram ($C = 1.96$). In this case GSA makes an error and overestimates the critical moment, which can be dangerous as the calculated load-bearing resistance of the structure is higher than it actually is.

In cases 3 and 4, SN003A (Bureau, 2005) is needed to determine the critical moment. The method which is provided here, applies for beams which are restraint against lateral movement and rotation about the longitudinal axis at each end. The elastic critical moment may then be calculated with the following formula. Equation 3.7 is applicable for straight members for which the cross-section is symmetric about the bending plane.

$$M_{cr} = C_1 \frac{\pi^2 EI_z}{(kL)^2} \left(\sqrt{\left(\frac{k}{k_w}\right)^2 \frac{I_w}{I_z} + \frac{(kL)^2 GI_t}{\pi^2 EI_z} + (C_2 z_g)^2} - C_2 z_g \right)$$

with :

- $E = \text{Young's modulus (N/mm}^2\text{)}$
- $I_z = \text{second moment of area about the weak axis (mm}^4\text{)}$
- $G = \text{shear modulus (N/mm}^2\text{)}$
- $I_t = \text{torsion constant (mm}^4\text{)}$
- $L = \text{beam length between points with lateral restraint (mm)}$
- $I_w = \text{warping constant (mm}^4\text{)}$
- $z_g = \text{distance between load application and shear centre (mm)}$

(3.7)

C_1 , C_2 , k and k_w are all constants depending on the loading and end restraint conditions. As in both example cases the support conditions at the ends are fork supports, k and k_w are both equal to 1. As the load is applied in the shear centre, C_2 is equal to 0. C_1 depends on the bending moment diagram and the same problems arise as in the earlier cases (see Table 3.2 in SN003A (Bureau, 2005)). It turns out Karamba again assumes a constant bending along the beam and thus underestimating the elastic critical moment to make a conservative assumption ($C_1 = 1$). The small dissimilarities in case 4 are caused by the small differences in the used E-modulus and cross-section properties in the different calculation methods.

When the calculation of the lateral torsional buckling resistance continues, both GSA and Karamba are following the Eurocode 3 (European Committee for Standardization, 2005) correctly. A more extensive elucidation of the calculations are shown in Chapter B.

Next, the results of Karamba and GSA of the analysis of a truss structure are compared. Figure 3.12 shows the truss, as modelled in Karamba.

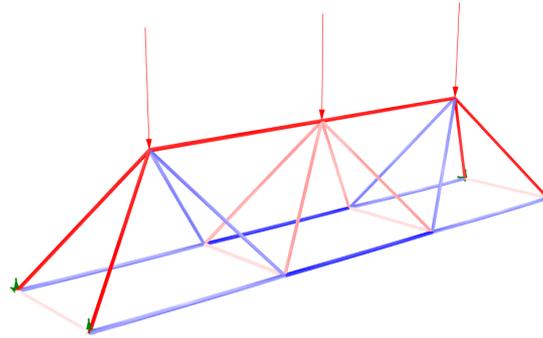


Figure 3.12: Karamba benchmark truss design

The truss is loaded by three point loads of 250 kN (as shown in Figure 3.12) and by gravity loading of all members. Each of these members are circular hollow sections. For this case again the stresses and maximum utilisation are compared. In Table 3.3 the utilisation and stresses are shown for this case, showing the results of both Karamba and GSA. Note that the stresses for Karamba are estimated from a colour legend, which has influence the accurateness.

Comparing the results, some values stand out. First of all, almost all results which are computed by Karamba are more conservative than the values calculated with GSA. This has to do with several reasons. GSA takes the plasticity into account in its calculation, which leads to some extra resistance and thus a smaller unity check. Although Karamba is able to take plasticity into account as well, it was chosen to make a conservative assumption and to let Karamba use the elastic values. On top of that, evaluating the axial utilisation, Karamba has already taken into account buckling, while GSA calculates the utilisation for different buckling types in separate unity checks. This explains the large differences between the analyses of the axial utilisation for compression members.

For the tension members, the differences are caused by something else. The results of Karamba and GSA for the axial utilisation of these members are almost equal. The overall utilisation however, differ significantly. The overall utilisation calculated by GSA is identical to the axial utilisation. However, Karamba adds the utilisation of the bending moments (in both directions) to the axial utilisation to come to the overall utilisation. Although it is a conservative assumption, it is not conventional. Despite the aforementioned differences between the two programmes, for this case none of the deviations in the overall utilisation are larger than 6%.

It can be concluded that Karamba was able to accurately analyse the structural 'performance' of most of the test cases. In the deviations and errors that did arise, Karamba is always conservative and will therefore not cause large problems in later stages of the design process. With the deviations and potential errors known beforehand, some values found by Karamba can be post processed and altered within Grasshopper to come to more representative results. Obviously, it is however very important to validate the found results, as not all possible cases are examined. The examples show that extra attention is especially needed for cases in which (lateral torsional) buckling is governing.

In Section 5.1, the structural objective and design variables for the case study are further explained.

3. Acoustic and structural analysis

Member	Overall utilisation (-)			Axial utilisation (-)		Axial stress (N/mm^2)	
	Karamba	GSA	Difference	Karamba	GSA	Karamba	GSA
1	0.027	0.020	0.7 %	0.000	0.000	+/- 8	+/- 7
2	0.032	0.022	1.0 %	0.004	0.002	+/- 8	+/- 9
3	0.032	0.022	1.0 %	0.004	0.002	+/- 8	+/- 9
4	0.027	0.020	0.7 %	0.000	0.000	+/- 8	+/- 7
5	0.140	0.087	5.3 %	0.091	0.087	+ 35	+ 38
6	0.200	0.150	5.0 %	0.150	0.150	+ 56	+ 54
7	0.140	0.087	5.3 %	0.091	0.087	+ 35	+ 38
8	0.970	0.940	3.0 %	0.910	0.240	- 77	- 78
9	0.970	0.940	3.0 %	0.910	0.240	- 77	- 78
10	0.140	0.087	5.3 %	0.091	0.087	+ 35	+ 38
11	0.200	0.150	5.0 %	0.150	0.150	+ 56	+ 54
12	0.140	0.087	5.3 %	0.091	0.087	+ 35	+ 38
13	0.960	0.930	3.0 %	0.860	0.260	- 88	- 88
14	0.310	0.300	1.0 %	0.280	0.083	- 33	- 31
15	0.130	0.095	3.5 %	0.100	0.095	+ 35	+ 34
16	0.960	0.930	3.0 %	0.860	0.260	- 88	- 88
17	0.310	0.300	1.0 %	0.280	0.083	- 33	- 31
18	0.130	0.095	3.5 %	0.100	0.095	+ 35	+ 34
19	0.130	0.095	3.5 %	0.100	0.095	+ 35	+ 34
20	0.310	0.300	1.0 %	0.280	0.083	- 33	- 31
21	0.960	0.930	3.0 %	0.860	0.260	- 88	- 88
22	0.130	0.095	3.5 %	0.100	0.095	+ 35	+ 34
23	0.310	0.300	1.0 %	0.280	0.083	- 33	- 31
24	0.960	0.930	3.0 %	0.860	0.260	- 88	- 88

Table 3.3: Comparison between structural analysis results of Karamba and GSA for benchmark truss structure

4

Optimisation approach and workflow

In this chapter, the workflow of the optimisation process will be further elaborated. Focus will be given on the overall multidisciplinary design optimisation approach, the MDO architecture, the optimisation algorithm and the post process. First of all, focus will be given on the Grasshopper plug-in which is used to control the optimisation process, Octopus.

4.1 Octopus

As stated earlier, Octopus is a plug-in for the Grasshopper application and applies evolutionary principles to parametric designs made in Grasshopper. It allows the search for multiple objectives at once, producing a set of optimised trade-off solutions between the extremes of each objective (Vierlinger, 2012). Octopus makes use of multi-objective evolutionary algorithms and its user interface is based on the user interface of Galapagos, the standard optimiser in Grasshopper. This means that it is relatively easy for users who are familiar with Grasshopper. It allows the user to change objectives during the search, choose preferred solutions and to save solution states (Vierlinger, 2012). A major advantage of working within Grasshopper, is the representation of the design in Rhinoceros. In that way the aesthetics of a design can be evaluated as well, which is very important in the AEC industry. On top of that it is free of charge, so usable by everybody.

Octopus needs several inputs from Grasshopper for the multi-objective optimisation. First of all it needs parameters/genes, which possible values will be explored by Octopus. These genes are connected to the Octopus component in red (Figure 4.1). The parametric model (in Figure 4.1 indicated as "Process") takes these parameters as input for the calculations and analyses. In this case, the parametric model describes the building design and does the structural and

4. Optimisation approach and workflow

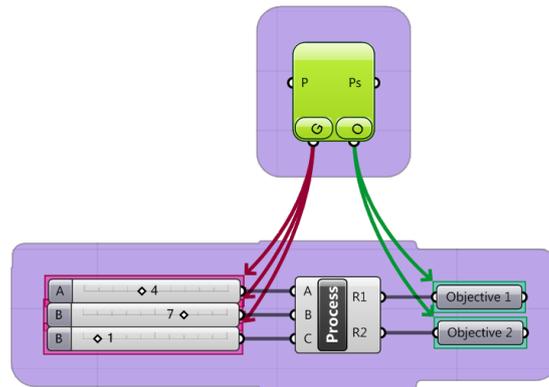


Figure 4.1: Example of Octopus set-up

acoustic assessment. Outputs of the model are the objectives, which will be evaluated by Octopus. These are connected with the green arrows. These objectives can be in the form of numeric values (which will be minimised by Octopus) and Booleans, which are hard constraints. If the Boolean is 'True', the solution is valid. Otherwise the solution will be dismissed. Octopus can also be given Meshes, which are 3D representations of the solution state and text objectives, which describe the objective dimensions (Vierlinger, 2012).

Other important settings that can be altered within the Octopus plug-in, are the optimisation algorithm (see Section 4.3) and inputs which the algorithms need such as the elitism, crossover and mutation rate and population size. Octopus will show the trade-offs between the different objectives as shown in Figure 4.2. Each block represents a design solution, which can be reinstated in Grasshopper.

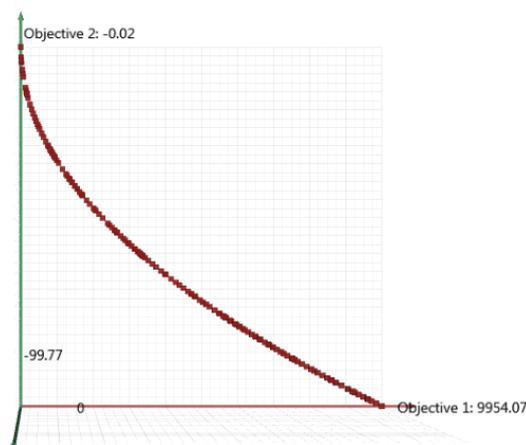


Figure 4.2: Example of a Pareto front output from Octopus

4.2 MDO approach and process

An important consideration when implementing multidisciplinary design optimisation is the organisation of the several discipline analysis models, approximation models and optimisation software in combination with the problem formulation. This combination is commonly referred to as MDO architecture. The choice of architecture, together with the chosen optimisation algorithm, can have a major influence on the computation time and the final solution (Martins and Lambe, 2013).

The MDO architecture describes (1) how the different disciplines (or systems) are coupled and (2) how the overall optimisation is solved. It can be monolithic or distributed. In monolithic approaches, a single optimisation problem is solved. Distributed approaches divide the problem in several 'sub'-problems, each with a small set of the variables and constraints (Martins and Lambe, 2013). This can for example be of use when parallel computing is desirable. The choice for the most appropriate MDO architecture is dependent on the human and computational organisation, the available algorithms and the design problem itself (Martins and Lambe, 2013).

To investigate the actual influence of the MDO architecture on the optimisation process, a simple case study is conducted. This case study is shown in Figure 4.3.

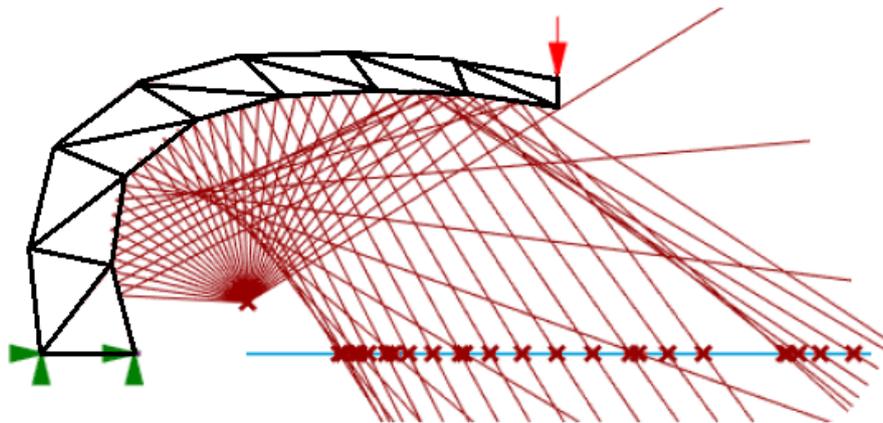


Figure 4.3: Design for MDO test case

This test case consists of a 2D cantilevering truss structure, loaded by its own weight and a point load. A sound source is located below this structure and omits sound towards the truss. This is reflected towards a line, which represents the receiver. Structural objective is to minimise the weight of the structure, while the displacements and stresses stay below the maximum allowable. The acoustic objective is to distribute the sound evenly across the line. The line is therefore divided in equal parts, which each should 'receive' an equal number of rays. Parameters that were explored, were parameters to describe the overall geometry and the cross-sections of the members. The geometry parameters are shared; they have an influence on both the structural and acoustic performance. The cross-sections only have an influence on the structural performance. The entire case is modelled in Grasshopper.

4. Optimisation approach and workflow

Several MDO architectures and Octopus set-ups are applied to this case. The results of these tests are shown in Figure 4.4. Note that the results found by Grasshopper and Octopus are exported to Excel with the help of the Lunchbox plug-in (Miller, 2012).

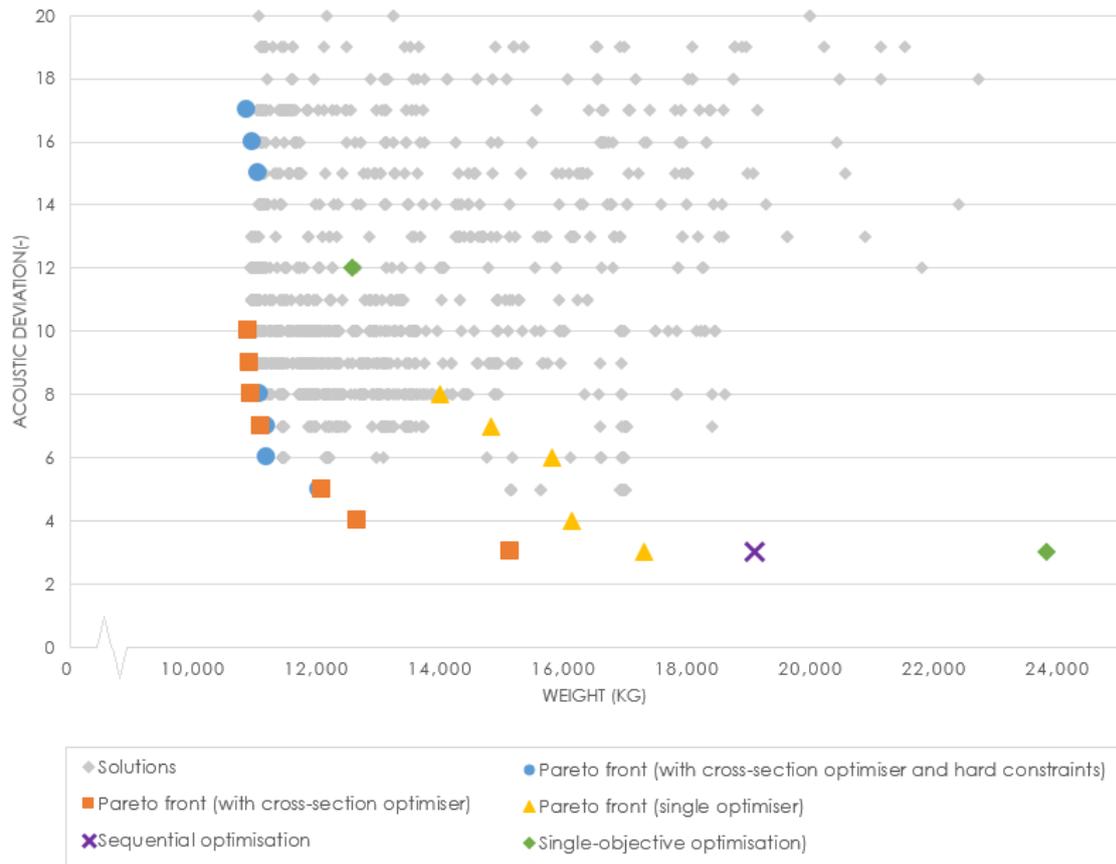


Figure 4.4: Results MDO set-up tests

The first set-up that was tested, used a single optimiser (yellow solutions). In this case all parameters are driven by Octopus. This however leads to a lot of infeasible and suboptimal designs, as the needed cross-sections are very dependent on the overall geometry. The optimal cross-sections are not always the smallest, as they will lead to an infeasible design in case the cross-section is chosen too small. To solve this issue, the Karamba cross-section optimiser (Section 3.2.2) was applied (orange solutions). This optimiser searches for the best cross-section from a given list of possible cross-sections. This means that the parameters describing the cross-sections are not driven by the overall optimiser. The overall optimiser decides the overall geometry, after which the cross-section optimiser searches for the optimal cross-section for that geometry. Figure 4.4 shows that this approach leads to better solutions than the single optimiser approach. A small adjustment leads towards another improvement in estimating the Pareto front. In the former approaches, infeasible designs are discarded by multiplying the structural performance indicator by a penalty factor. By giving Octopus a hard Boolean constraint (blue solutions), Octopus discards the infeasible designs itself. This leads towards better exploration at the 'structural end' of the Pareto front. In contrast, the penalty factor causes

4. Optimisation approach and workflow

a shift from these ends, as Octopus receives very bad performances and stays clear from this unexplored design space. However solutions on the edge of feasibility have very good structural performance and need to be incorporated in the estimation of the Pareto front. Note that some of the blue solutions disappear below the orange ones.

To compare the multidisciplinary design optimisation with more traditional optimisation approaches, also a sequential optimisation is applied with Grasshopper's own single-objective optimiser Galapagos (purple solution). With all parameters involved, the design is optimised with respect to the acoustic objective. In the next step, just the cross-sections are optimised to minimise the weight. The other parameters cannot be altered, as that would change the acoustic performance. Single-objective solutions are shown in green. One solution is structurally optimised and the other is just optimised for the acoustics. In this case, all these traditional optimisation approach did not find a solution close to the Pareto front. Although these approaches should be able to find a solution on the Pareto front, the biggest drawback is that these approaches do not give any insight in the trade-offs between the different disciplines.

The described approach, with a separate cross-section optimiser and hard constraints, will be applied on the stadium roof case study. In Figure 4.5 the chosen approach is schematised. The overall optimiser drives the parameters which describe the overall geometry of the roof (z_{sh}) and structure (z_2) towards minimal structural weight (f_2) and minimal sound pressure level deviations (f_1). The structural optimiser searches for the optimal cross-sections (x_2) for the found geometries.

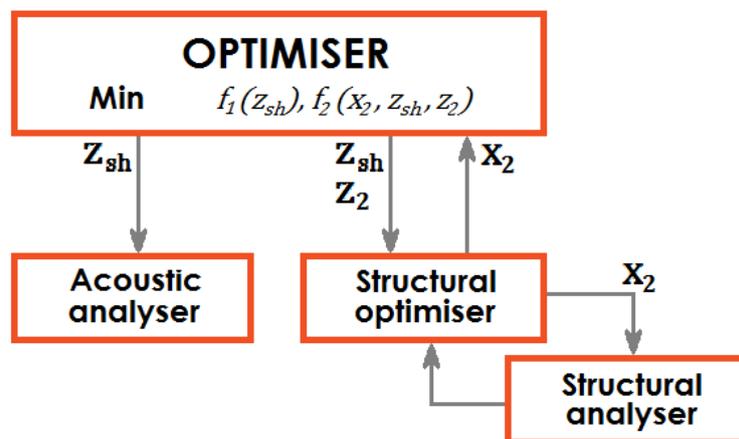


Figure 4.5: Schematisation MDO architecture

Figure 4.6 illustrates the overall process, using the Design Structure Matrix (DSM). In this diagram, the computational optimisation, parametric modelling and performance assessment modules are represented by cells and in sequence from top-left to bottom-right. Horizontal lines represent cell outputs and vertical lines represent inputs for a cell.

Outputs from the optimiser (Octopus) cell are the design variables, which describe the parametric model (Rhino + Grasshopper). This model describes the geometry. The performance of the geometry needs to be assessed for both acoustics and structure. The acoustic ana-

4. Optimisation approach and workflow

lysis module assesses the acoustic performance and feeds this back to the optimiser (custom ray tracer). The structural assessment first searches for the optimal cross-sections for the given geometry, after which the structural performance is guided back to the overall optimiser (Karamba). The objective functions and constraints of the optimisation problem are evaluated by the optimiser to investigate the trade-offs.

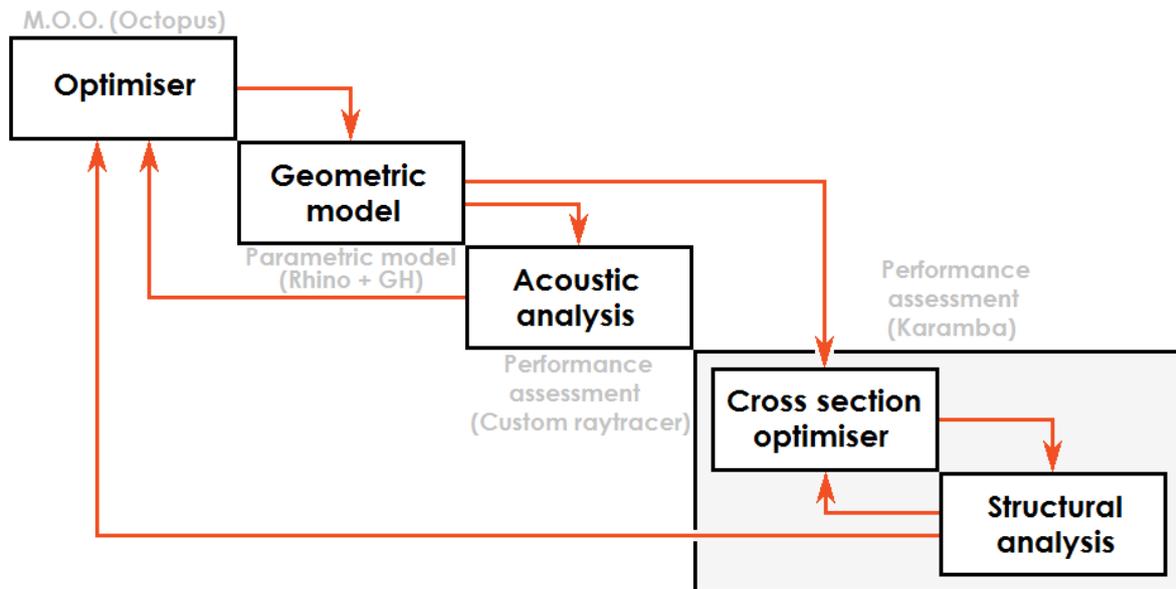


Figure 4.6: Applied Design Structure Matrix

4.3 SPEA-2 and HypE

Octopus is based on SPEA-2 and HypE from ETH Zurich and the user interface of David Rutten's Galapagos (Vierlinger, 2012). SPEA-2 and HypE are both multi-objective genetic algorithms which Octopus uses to investigate the design space. Both algorithms will be briefly described in this section.

SPEA-2 is an improved version of the in 1999 Strength Pareto Evolutionary Algorithm, which is a technique for finding or approximating the Pareto-optimal set (Zitzler et al., 2001). SPEA was one of the first algorithms to apply elitism (Section 2.3), which showed great value in multi-objective search. SPEA uses a regular population and an external archive. In the first step, all non-dominated members of the population are copied to the archive and the (now) dominated or duplicate solutions are removed from the archive. If the size of the archive exceeds a predefined limit, a clustering technique deletes certain archive members. This technique preserves the characteristics of the non-dominated front (Zitzler et al., 2001). In the next step, fitness values are assigned to both the population and archive members. This is done as follows (Zitzler et al., 2001).

4. Optimisation approach and workflow

- Each individual in the archive is assigned a strength value, which represents its fitness and is dependent on the number of population members that are dominated by the individual.
- Each individual in the population is assigned a fitness value by summing the strength values of the archive members which dominate the individual.

During the mating selection, binary tournaments are used to select members from both the archive and population. SPEA however shows several weaknesses (Zitzler et al., 2001). The fitness assignment leads to problems when the archive contains only one member; in that case the entire population is assigned the same fitness value. Density information is only known for archive members. This causes problems when many individuals do not dominate each other. On top of that, the clustering technique which is used, may lose outer solutions of the archive.

SPEA-2 solves these issues, by altering some characteristics of the SPEA algorithm. First of all, the fitness assignment is based on both how many individuals a solution dominates and by how many individuals it is dominated by. Secondly, a nearest neighbour density estimation allows a more precise guidance of the search process. Finally, an improved reduction method preserves boundary solutions of the archive (Zitzler et al., 2001). Drawback of these improvements are the computationally costs that they come with (Konak et al., 2006).

HypE is a hypervolume estimation algorithm, developed at the ETH Zurich. The hypervolume indicator is the only set measure that is known to assess the quality of a Pareto set approximation (Bader and Zitzler, 2008). This means that whenever a Pareto set approximation entirely dominates another, the hypervolume indicator will be better as well. This property is of great relevance for problems involving a large number of objective functions (Bader and Zitzler, 2008). It however does require great computational power for the calculation. HypE is able to make a trade-off between the accuracy of the hypervolume estimation and available computing resources. Experimental results indicate HypE is highly effective for many-objective problems in comparison with other multi-objective evolutionary algorithms (Bader and Zitzler, 2008).

4.4 Post processing

After the optimisation process explored the design space, assessed the performances of the different solutions and estimated the Pareto front, it can be difficult to interpret the optimisation results and find the actual best fitting solution for the design problem. This post processing of the result is important to reduce the complexity of the optimisation problem and providing improved designs (Yang et al., 2015).

Grasshopper plug-in Octopus makes use of a scatter plot to show the trade offs between the different objectives (see Figure 4.2). Each solution is represented by a block and can be reinstated in Grasshopper. When an optimisation problem only has two or three objectives, it is easy to find and reinstate solutions with a certain performance and to see their corresponding design. However, the scatter plot becomes cluttered when a problem has more than three objectives as the plot needs more than 3 dimensions to show all performances. In these cases, Octopus shows the scatter plot by changing the colours and sizes of the points in the plot, but

4. Optimisation approach and workflow

it is very difficult if not impossible to investigate the exact trade offs between the different objectives. Consequently, it is very difficult to find the desired result and design when the amount of solutions increases.

An other method of visualising the results, is by means of a parallel coordinate chart. In these charts all 'dimensions' are represented by equally spaced vertical lines. This means that not only objectives, but also variables can be plotted to get even more understanding in the optimisation problem and its results. Each solution is represented by a polyline with vertices on each vertical line, representing its value in that 'dimension'. These charts tend to stay more clear than scatter plots in case of higher dimensional optimisation problems. Despite the fact that Octopus has a 'multi-axes view', it is not a useful one for post processing the results.

For that reason, it was chosen to develop a custom post processing tool, named PPOR (Post Processing of Optimisation Results). This tool will be further explained in the next section. Figure 4.7 shows an example of a parallel coordinate chart (or parallel plot), applied for a test case which minimises the ground area of a box, whilst maximizing its volume. One of the solutions is shown in green as an example.

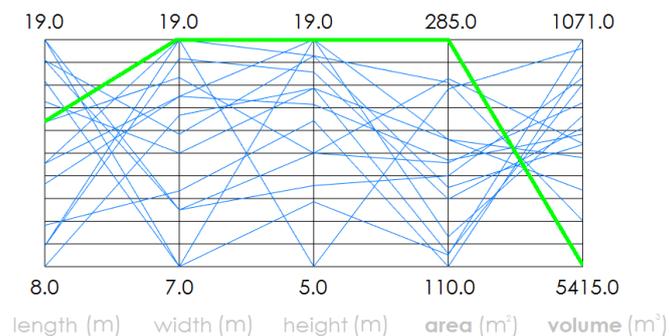


Figure 4.7: Example of a parallel coordinate chart

4.4.1 Post processing tool

To be able to assess the found solutions after the optimisation process, a custom post processing tool (PPOR) for Grasshopper was developed. The most important input for the tool are the variables, objectives and other performance indicators which are needed for the assessment and need to be shown in the parallel plot. Obviously, the variables and objectives used for the optimisation are necessary, but it could be useful to add other performance indicators as well. These could then be used to filter out found solutions which prove to be insufficient. Hard constraints are formulated beforehand and guide the optimisation process. However, by using PPOR as proposed, more solutions are evaluated during the optimisation and can be filtered out subsequently. This is an useful approach in case constraints are negotiable. By applying hard constraints, focus is given on finding solutions as close to to the true Pareto front as possible. This is in expense of finding the extreme solutions at the ends of the Pareto front, for which the post process method is superior. Therefor, the best suited approach is dependent on the constraint and the goal of the optimisation process; design optimisation or design exploration.

4. Optimisation approach and workflow

Another important input for the tool, is a representation of the design. In that way, the appearance of a building could also be evaluated during the post process. Finally, two indices representing the variables or objectives which need to be shown in a scatter plot are to be entered. In case a optimisation process only includes two objectives, a scatter plot showing the trade offs between these two objectives is also shown. Optional inputs are the indices of the variables, objectives or other performance indicators together with a range which should include their values. These ranges act as the earlier mentioned filters. Appendix C shows the Grasshopper component.

Figure 4.8 shows the outcome of PPOR for the box test case as seen in Rhinoceros, with a filter applied on the box volume. The selected solution is shown as a green point in both graphs, as a green polyline in the parallel plot and a representation of the design is shown in the bottom left corner (top view). Orange solutions are Pareto optimal solutions and the selected range (2000-5414) is shown in yellow. Furthermore, Grasshopper shows the parameter values of the selected solution and indicates if it is a Pareto optimal solution or not.

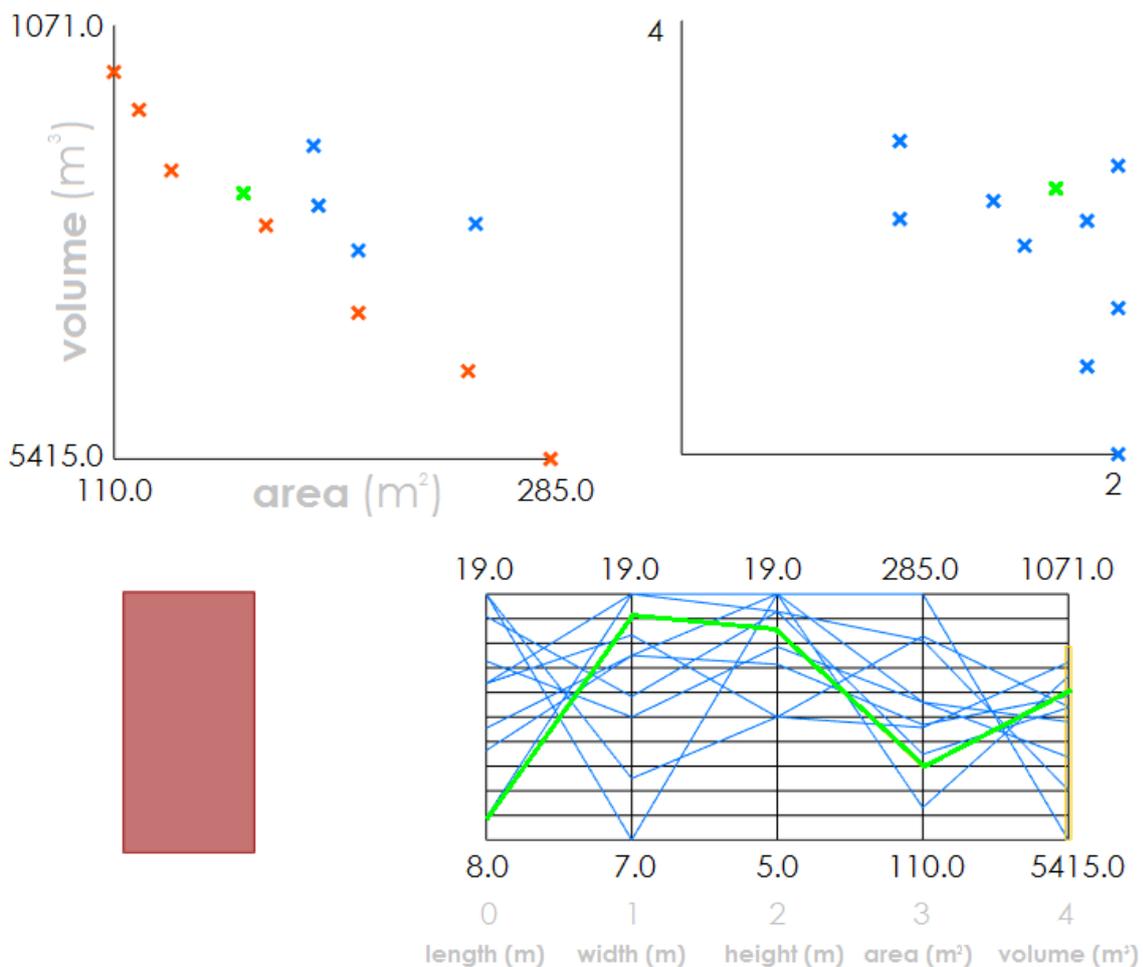


Figure 4.8: Output produced by the custom post processing tool PPOR

4. Optimisation approach and workflow

The functioning of PPOR is mainly based on the 'Data recorder'-component of Grasshopper. These recorders need to be put in between the parameters and the tool. During the optimisation process, Octopus alters the variables and the values of the objectives and/or performance indicators are calculated. The data recorders save all these values for each evaluated solution. In that way, they contain a list of all values when the optimisation is stopped. These are then entered into PPOR. The infeasible designs, which do not satisfy the constraints, are removed from the value lists by a culling pattern.

Validation of the post processing tool

Before using PPOR in the case study, it should be investigated whether or not the tool operates as it should. It was chosen to model a optimisation problem as found in literature, named the cone problem (Marcuzzi, 2009), in which both the base area and the lateral surface area of a cone needs to be minimised while the volume needs be above a certain value. The problem can be described as shown in Equation 4.1.

$$\begin{aligned} & \min S \\ & \min T \\ \\ \text{with : } & S = \pi \cdot r \cdot s \text{ (base area)} \\ & T = \pi \cdot r \cdot (r + s) \text{ (lateral surface area)} \\ & s = \sqrt{r^2 + h^2} \\ & r = \text{base radius (0 - 10)} \\ & h = \text{height (0 - 20)} \\ \\ \text{and : } & V > 200 \\ \text{with : } & V = \frac{\pi}{3} \cdot r^2 h \end{aligned} \tag{4.1}$$

The most important check is to compare the Pareto front as shown by PPOR with the Pareto front found by Octopus (after around 3.600 evaluated solutions). This comparison is shown in Figure 4.9 and demonstrates that the 22 found Pareto solutions are identical in both cases. This is a good check to perform each time PPOR is applied. The graph at the right shows the Pareto front as found in the literature, with the points as found by PPOR in it as well.

PPOR was further validated by comparing the performances of 10 random solutions as shown in the tool with the recalculated performances. These results are shown in Table 4.1. From these results it can be concluded that the tool operates adequately.

4. Optimisation approach and workflow

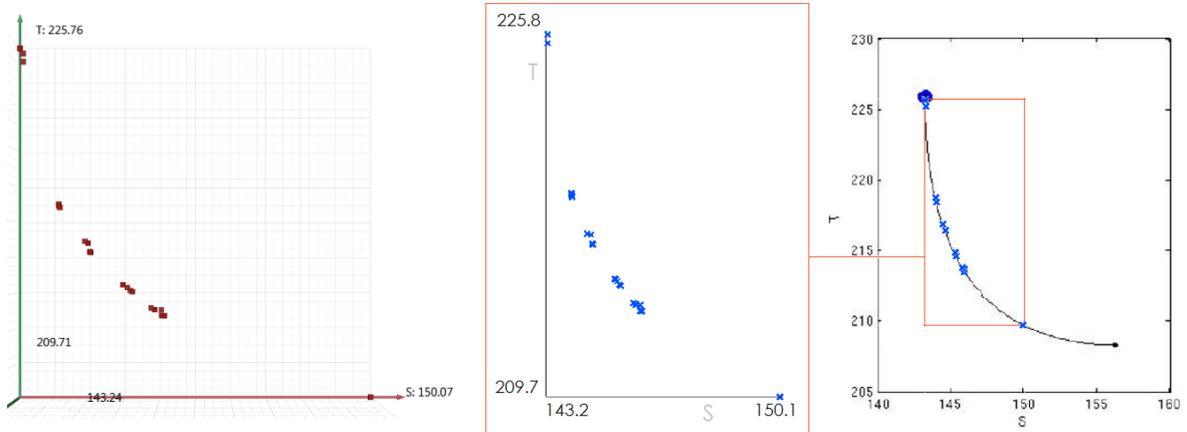


Figure 4.9: Comparison of the results in Octopus (left), the custom post processing tool (middle) and literature including the solutions from PPOR (right)

		Tool results		Calculation		Differences	
h	r	S	T	S	T	S	T
10.40	4.30	152.03	210.12	152.03	210.12	0.0 %	0.0 %
14.58	8.39	443.39	664.53	443.39	664.53	0.0 %	0.0 %
9.53	5.35	183.62	273.44	183.62	273.44	0.0 %	0.0 %
8.57	8.14	309.83	517.99	309.83	517.99	0.0 %	0.0 %
12.51	4.33	180.27	239.29	180.28	239.29	0.0 %	0.0 %
8.57	5.14	161.38	244.35	161.38	244.35	0.0 %	0.0 %
7.37	5.32	151.97	240.88	151.97	240.88	0.0 %	0.0 %
14.87	6.95	358.14	509.76	358.14	509.76	0.0 %	0.0 %
10.67	6.68	264.27	404.49	264.27	404.50	0.0 %	0.0 %
12.84	9.88	502.56	809.04	502.56	809.04	0.0 %	0.0 %

Table 4.1: Validation of 10 random solutions from the post processing tool PPOR

4. Optimisation approach and workflow

4.5 Further possible applications

One of the main starting points, deciding on a multidisciplinary design optimisation approach, was that the approach should be applicable for other design problems and disciplines. It should not be solely focused on the case study in this research or limited to the two disciplines that are investigated in this research, structure and acoustics. However, as stated earlier, (most) buildings are one-of-a-kind designs. This means that they have a unique problem statement, which makes the standardisation of a solution strategy difficult. For different designs, different disciplines are decisive and the interactions between different disciplines are different per building, much less 'organised' than in designs in other engineering fields (Jansen et al., 2014). In this section, it is argued if the described approach is applicable for other problems and if so, in what manner.

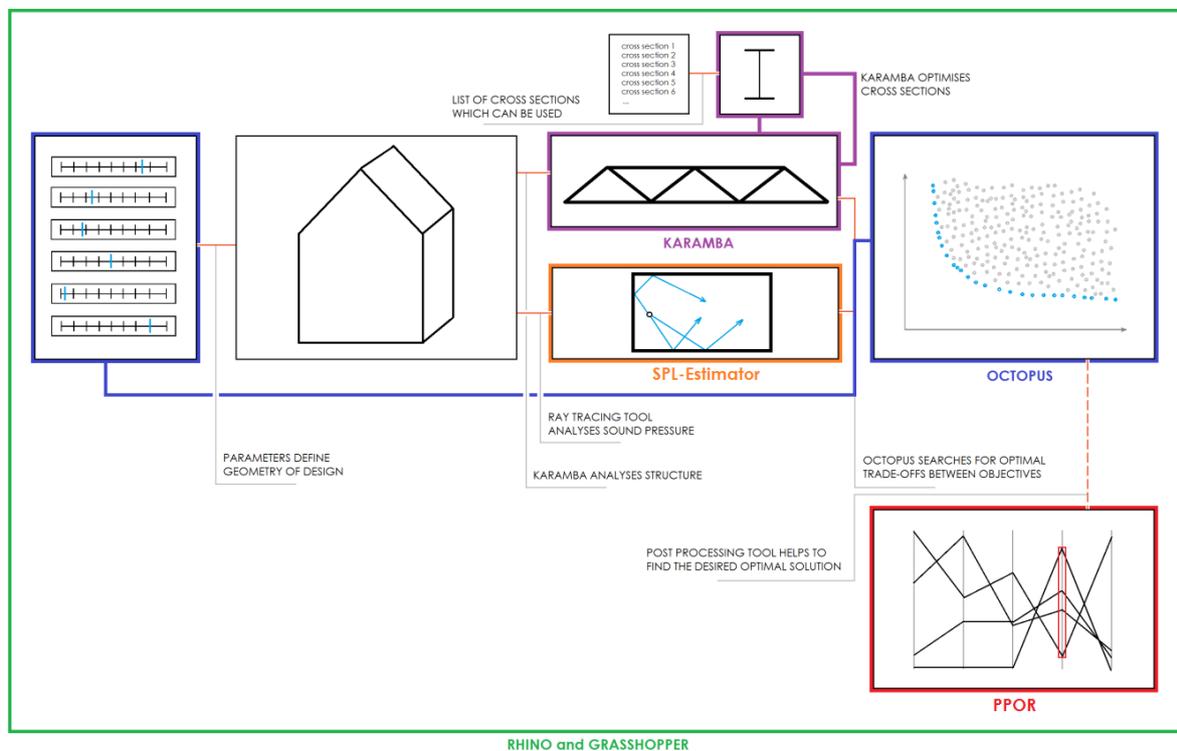


Figure 4.10: Schematic overview of the proposed MDO approach

Figure 4.10 shows a schematic overview of the approach as applied in this research. Outputs from the optimiser are the design variables, which describe the geometry in Grasshopper). The performance of the geometry then is assessed for the different disciplines. In this case, these disciplines are acoustics and structure. The acoustic analysis module assesses the acoustic performance and feeds this back to the optimiser (custom ray tracer). The structural assessment first searches for the optimal cross-sections for the given geometry, after which the structural performance is guided back to the overall optimiser (Karamba). The objective functions and constraints of the optimisation problem are then evaluated by the optimiser to investigate the trade-offs. The assessment 'blocks' (in this case the SPL-Estimator and Karamba) could however be substituted by other analysis programs for other disciplines. In the earlier described

4. Optimisation approach and workflow

cone problem (Section 4.4.1) for example, the analysis 'blocks' consisted out of simple formulas evaluated by Grasshopper itself. This shows the approach is applicable for other (design) problems.

However, there are some important requirements for applying the approach. It is possible to let programs outside Grasshopper analyse the geometry for a certain discipline, but the assessed performance indicator must be fed back into Grasshopper for Octopus to evaluate the objective function(s). Furthermore, the ideal MDO architecture is dependent on the optimisation problem (Section 2.2). Thus, the way the optimisation and analysis is organised and how the different constraints, variables and objective functions are handled should be investigated for each design problem. Consequently, sufficient knowledge about this and about Grasshopper, as the entire approach is set-up within Grasshopper, are needed for someone to apply the proposed approach.

5

Case study

This chapter will illustrate the chosen case study used for testing the described multidisciplinary design optimisation approach. As stated earlier, focus will be given on the structural and acoustic performance of a large span steel stadium roof. Firstly, the design problem is outlined. The chosen stadium roof is defined, together with the objectives, variables and constraints (see Section 2.1) of the optimisation process. Next, the modelling approach is explained, together with the made simplifications for and the limitations of the model.



Figure 5.1: Picture of Donbass Arena (FC Shaktar Donetsk, 2015)

5. Case study

5.1 Design problem

It was chosen to use the 'Donbass Arena' as case study to apply the proposed multidisciplinary design optimisation approach on. The Donbass Arena is situated in Donetsk, Ukraine and was designed by Arup Sport. It opened in August 2009 and is home to the Ukraine football club Shaktar Donetsk. It hosted several UEFA Euro 2012 matches and is also used for concerts. The stadium was damaged several times in 2014 during the conflicts in Ukraine, after which Shaktar provisionally moved its home games to Lviv (Wikipedia, 2016c). Table 5.1 shows some important Donbass Arena facts, which are used to set up the model (FC Shaktar Donetsk, 2015).

Field size	105m x 68m
Capacity	52.187
Covered capacity	93% of stands
Height from pitch to roof	54m
Mass of all steel used in roof	3.800 tonnes

Table 5.1: Donbass Arena facts

5.1.1 Stadium roof

The optimisation process is mainly focussed on the steel roof of the stadium to reduce the size and complexity of the design problem. The roof was chosen as it has a large influence on the acoustics inside the stadium and the needed span of the roof, causes a structural challenge. The Donbass Arena has a sloping roof, which consists of a network of steel elements. On top of these elements, the biggest part of the roof is cladded by metal sheets. A small part is translucent, which ensures sufficient daylight reaches the grass of the pitch. The cladding is the main sound reflecting surface of the roof and changing the geometry of the roof will lead to different reflections of the sound. The size of the 'opening' in the roof has a large influence on the amount of lost sound.

The roof is carried by 12 main cantilevering trusses, of which 4 are twice as big as the other 8. These trusses carry the rest of the steel elements and are divided around the elliptic perimeter of the roof. The structural elements during the construction phase of the roof are shown in Figure 5.2. It should be noted that the elements of the 12 cantilevering trusses are considerably bigger than the other elements. This, in combination with the connections between the cantilevers and the other elements, indicates that there is no compression ring at top of the roof. This is very important for the modelling of the structure, as a compression ring would lead to a different structural system.

In the following sections, explanation is given about the modelling and set-up of the optimisation process of the stadium and its roof.



Figure 5.2: Roof structure of Donbass Arena during construction (Academic, 2014)

5.1.2 Objectives

Multi-objective genetic algorithms are able to optimise several fitness functions, by altering design variables (Echenagucia et al., 2014). These fitness functions (or objective functions) and design variables have to be well formulated by the designers for the optimisation algorithm to properly search the design space and to find the desired optimal solutions.

Acoustic objective

When investigating the main acoustical difficulties faced in a stadium, some key problems arise. The first important acoustical “feature” of a stadium, is its atmosphere. A good atmosphere at a sports stadium is important for both the players and the crowd. As sound loses energy by travelling, the key variable is the volume of the venue (Owens, 2014). Another main variable is the positioning of reflecting surfaces, as hard surfaces are able to reflect noise back to the crowd.

The shape of the stadium is also important when looking at another problem. Sound in stadiums can be very localised (Vennard, 2013). As acoustics is about geometry (see Section 3.1.1), there can be parts in stadiums where it is impossible to see, and thus possibly difficult to hear, what is going on in other parts of the stadium.

Finally, another issue faced with stadium design is its surroundings. Especially with stadiums in urban areas, nuisance for its surroundings could be an important influence for the design. Note that these aspects are important when a stadium is used for sporting events and when it is used for other events, such as concerts. Although there are numerous other aspects which can be important for the acoustical design, focus was given on these general aspects during the optimisation process. When combined, it could be described as the search for a uniform distribution of sound energy among all listening positions (Echenagucia et al., 2014). This leads to the objective function as described in Equation 5.1.

5. Case study

$$\begin{aligned} \text{minimise} \quad & E_{tot} = \sum |SPL_r - SPL_i| \\ \text{with :} \quad & SPL_r = 10 \log \left(\frac{P_s Q}{RAI_0} \right) \\ \text{and :} \quad & SPL_r = \text{ideal SPL for receiver (dB)} \\ & SPL_i = \text{calculated SPL for } i^{\text{th}} \text{ receiver (dB)} \\ & P_s = \text{source sound power (W)} \\ & Q = \text{directivity factor} \\ & \quad (= 1 \text{ for omnidirectional}) \\ & R = \text{total number of receivers} \\ & A = \text{average area of a receiver surface (m}^2\text{)} \\ & I_0 = \text{reference sound intensity} = 10^{-12} \text{ W/m}^2 \end{aligned} \tag{5.1}$$

E_{tot} is the total summed 'error' for the sound pressure levels for all listening positions. Minimising this will lead to a search for a stadium where are listening have an SPL as closely to the target value as possible. This means the number of rays escaping the stadium bowl (nuisance) and the localisation of sound is minimised, as rays are guided by the optimisation process towards listening positions (stands and field). The exact individual sound pressure levels are not essential, as long as the deviations of the estimations are not excessively large and the overall problem areas of the building can be distinguished. Further explanation of the acoustic optimisation approach, see Section 3.1.

Structural objective

As stated earlier, the size of a stadium and the required span of its roof result in a large amount of material needed for the structure. Minimising this amount could have both environmental and economical advantages. The search for the 'optimal' structural design depends heavily on its objective. In some cases costs for the structure is the main performance indicator, in others it is its efficiency or mass.

In this case, minimisation of the structural mass of the roof was chosen as objective function. Karamba is capable of determining the sum of the mass of all modelled structural elements. Although the amount of material (and thus its weight) is an important influence on the costs for the structure, it is not the only one. Manufacturing and construction also greatly affect the costs for the structure and are both determined by for example the chosen type of sections and repetition of elements and connections. This means that the use of a certain type of section may lead to the lightest structure, but not the cheapest. Therefore it should be noted that the optimal structure in terms of mass is not necessarily optimal in terms of costs. The structural objective can be described as in Equation 5.2.

$$\begin{aligned}
 \text{minimise} \quad m &= \rho \sum A_i \cdot l_i \\
 \text{with :} \quad m &= \text{mass (kg)} \\
 \rho &= \text{mass density (kg/m}^3\text{)} \\
 A_i &= \text{area of cross – section of element } i \text{ (m}^2\text{)} \\
 l_i &= \text{length of element } i \text{ (m)}
 \end{aligned} \tag{5.2}$$

5.1.3 Variables

In this section, the variables which are altered during the optimisation process, are illustrated. Their influences on both objectives are investigated by means of a sensitivity analysis. The variables, their range, starting value (the values which describe the actual design of the Donbass Arena) and step size are shown in Table 5.2. In Section 2.1, more explanation is given on design variables.

	Lower limit	Base-line	Upper limit	Step size
0. Size of the roof 'opening' (m)	35	64	105	1
1. Mean height difference between roof edges (m)	-10	12	15	1
2. Shift of sine function describing 'inside' ellipse (-)	0	0	1	0.1
3. Extremes in sine function describing 'inside' ellipse (-)	0	0	20	2
4. Amplitude of sine function describing 'inside' ellipse (m)	0	0	5	0.1
5. Shift of sine function describing 'outside' ellipse (-)	0	0	1	0.1
6. Extremes in sine function describing 'outside' ellipse (-)	0	0	20	2
7. Amplitude of sine function describing 'outside' ellipse (m)	0	0	5	0.1
8. Radial divisions of cantilevering trusses (-)	10	14	20	1
9. Structural height at beginning of cantilevering trusses (m)	1	6	10	1
10. Structural height at end of cantilevering trusses (m)	1	3	10	1

Table 5.2: Variables of the Donbass Arena case study

In Figure 5.3, the variables are indicated in a schematic view of the (modified) Donbass Arena. Variable 1 describes the opening in the roof, it is the length of the major axis of an ellipse. The minor axis is determined by dividing the major axis by 1.2. Variable 2 describes the mean height of the roof at the ends of the cantilevering trusses, relative to the other end of the trusses. Variables 3, 4 and 5 together describe the shape of the 'inside' ellipse. To alter the shape of the roof, it was chosen to describe the height of this ellipse with a sine function. This sine function has a number of extremes, with a certain amplitude and which can be shifted around the ellipse. Variables 6, 7 and 8 do the same for the 'outside' ellipse. Variables 9, 10 and

5. Case study

11 describe the cantilevering trusses. The divisions in the trusses can be altered, together with the structural height at both ends. Note that all cantilevers have the same number of divisions and structural heights.

Lastly, the cross-sections of the structural elements are chosen as state variables and are driven by the separate cross-section optimiser (Section 3.2.2, Section 4.2 and Section 5.2.2). This cross-section optimiser has the choice of 455 different possible cross-section, starting from CHS219.1x6.3.

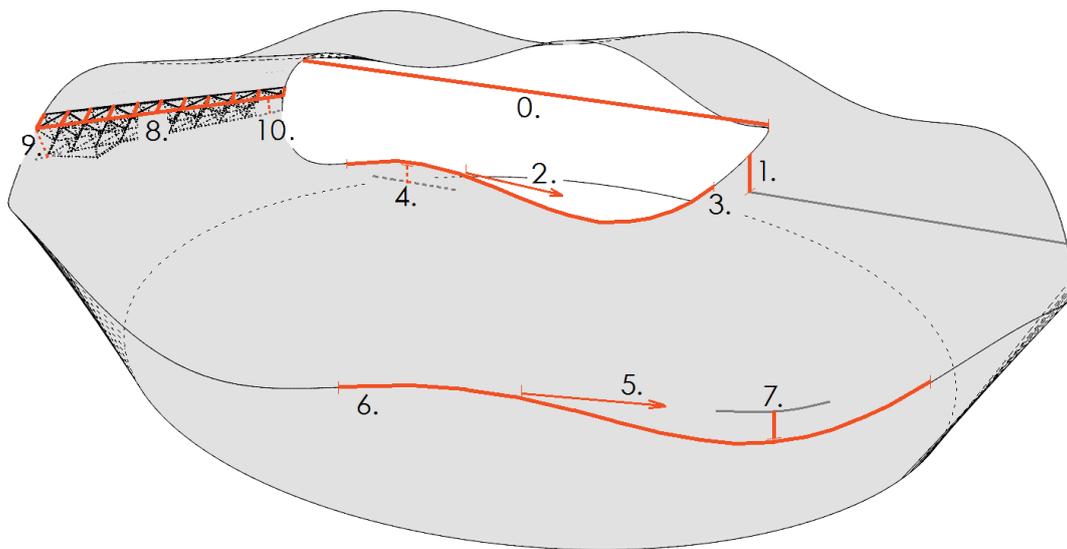


Figure 5.3: Variables describing the Donbass Arena

Sensitivity analysis

To study the influence of the different variables on both objectives, a sensitivity analysis was carried out. This analysis not only helps to increase knowledge about the different influences, but also helps checking the robustness of the model. It was chosen to use an one-at-a-time approach, which is the most straight forward approach. In this approach, only one variable is changed at a time to investigate its effect on the different performances. Each variable is altered to its minimum and maximum value, while the others are kept at baseline. By altering the variables one by one, each observed change in the outcome must be caused by that variable.

This approach however does not explore the input space entirely, as it does not take interactions between different variables into account. These interactions are especially important for variable 'groups' 3, 4 and 5 and 6, 7 and 8 and that is why the sensitivity analysis does not tell much about these variables. During the analysis of each of these variables, the remaining variables of the 'group' are set to the half of the range as they otherwise do not have an influence on the outcome.

Figure 5.4 shows the outcome of the analysis. For each variable the influence on both the structural objective (orange lines) and the acoustic objective (blue lines) are investigated. Their values are altered to their lower limit (red) and upper limit (green) one by one, after which the deviations of the objectives are determined.

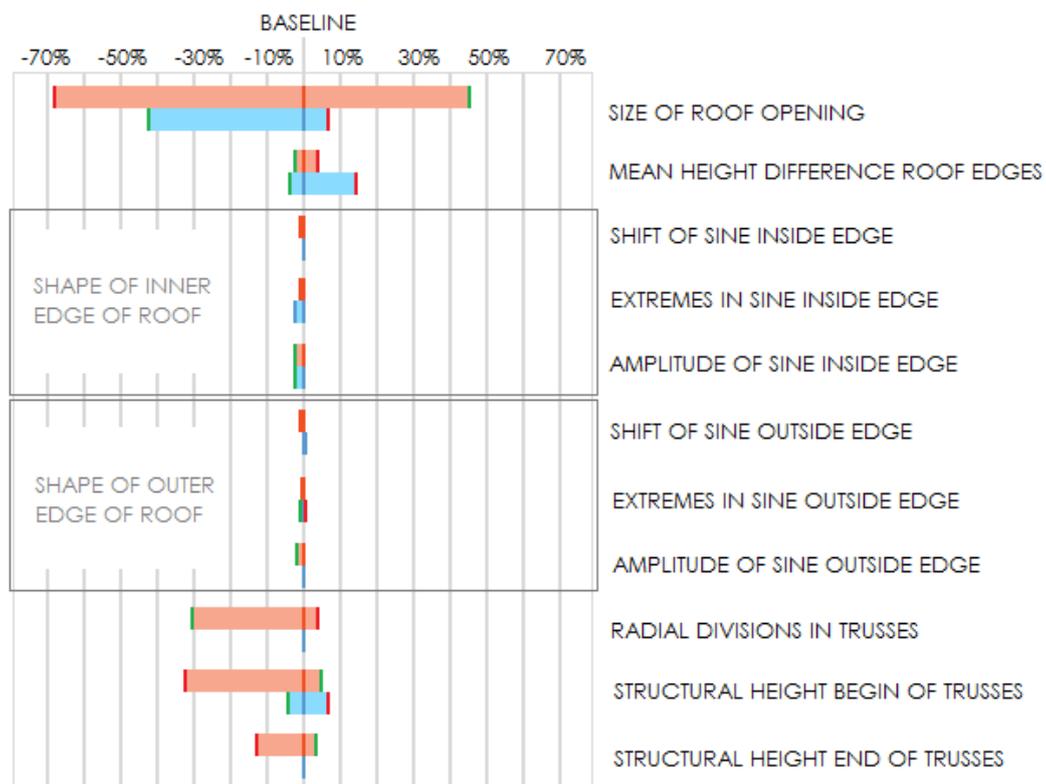


Figure 5.4: Sensitivity analysis of the different variables

As can be seen in Figure 5.4, the size of the roof opening (1) has the biggest influence on both the structural and acoustic performance. Increasing its size to the upper limit (105, see Table 5.2) leads to an improvement in the structural performance of almost 50% (the acoustic performance is decreased by over 40% due to sound loss), as the span of the trusses and thus the mass decreases. On the contrary, decreasing the opening size to the lower limit improves the acoustic performance by almost 10% and decreases the structural performance by 70%. Notable for the influence of the mean height at the end of cantilevering trusses (2) is that lowering this value both improves the acoustic and structural performance. This is probably caused by a small decrease of the cantilever spans (smaller structural mass) and a shorter travelling distance by the ray (less sound loss).

As stated earlier, the analysis does not tell much about variables 3 to 8, as they largely depend on each other. As, expected the number of divisions in the cantilevering trusses (9) do not influence the acoustics. The lower limit has a positive influence on the structural performance in this case, although the it is expected that the optimal number of divisions is largely dependent on the span of the trusses. Increasing the structural height at both the beginning and the end of the cantilevering trusses (10 and 11), improves the structural performance according to the

5. Case study

analysis. However, also for these variables applies that their optimal largely depends on other variables. Increasing the structural height at the beginning of the trusses has a negative influence on the acoustics, as it enlarges the volume of the stadium (height of the roof increases with the structural height). At the end of the trusses, the structural height does not influence the overall height of the roof and thus has no influence on the acoustics.

As the outcome of the sensitivity analysis can be explained logically for the variables that are not too dependent on other, the model of the Arena seems to be set up correctly. In Section 5.2, this is explored and elaborated further. In Chapter 6 and Chapter 7, a more elaborate investigation of the results of the optimisation process together with the variable influences is given. In Section 5.3, the baseline design is specified.

5.1.4 Constraints

In the search for the best performing designs, the design has to meet some requirements. These hard constraints are formulated beforehand and guide the optimisation process. An example is the structural utilisation of an element. However, 'constraints' could also be introduced in the post process (Section 4.4). In that way, possible solutions are evaluated during the optimisation process but can be filtered out subsequently. This is an useful approach in case constraints are negotiable. In case of a stadium, an example could be the capacity. If a solution performs significantly better with a slightly lower than desired capacity, it could still be an interesting solution.

It was chosen to apply both methods in this case study to explore the design space comprehensively and to be able to evaluate both approaches. The first 'hard constraints', which are given as input for the optimisation process, are the utilisation factors of all elements. These should be smaller than 1.0 for the action effect on the element to not exceed the element resistance (European Committee for Standardization, 2005). Note that the design actions need to be determined in the Ultimate Limit State (ULS), which is the limit state that concerns the safety and integrity of the structure.

$$\text{unity check} = \frac{E_{d,i}}{R_{d,i}} \leq 1.0 \quad (5.3)$$

with : $E_{d,i}$ = effect of design actions on element i
 $R_{d,i}$ = design resistance of element i

The deflection of the structure should be smaller than the maximal allowable limit given by the Eurocodes (Nederlands Normalisatie-Instituut, 2011). In this case, the actions on the structure need to be determined in the Serviceability Limit State (SLS), which is the limit state that concerns the functionality of the structure under normal use. For cantilevers, the following constraint applies (Equation 5.4).

$$\text{maximum deflection} \leq \frac{l_{rep}}{250} \quad (5.4)$$

$$\text{with : } l_{rep} = 2 \cdot \text{length of the cantilevers}$$

Finally, spectators should be able to have a clear view of the entire field. This means that the roof should not block their sight lines. Solutions which do block the view of some spectators, should therefor be indicated as infeasible.

$$\text{number of sightlines blocked by structure} = 0 \quad (5.5)$$

Two other values are determined to be able to apply extra constraints during the post processing. First of all the percentage of spectators covered by the roof is calculated. Secondly, the percentage of lost sound through the roof opening is determined. These values can be used to filter out undesirable solutions after the optimisation process. Note that the variables and objectives could also be used to filter solutions (see Section 4.4).

5.2 Modelling

In this section, the Grasshopper model representing the Donbass Arena, which is used in the optimisation process, will be described. The model is divided according its three main aspects, namely the part describing the overall geometry, the structural part of the model and the acoustic part of the model. These parts will all be illustrated, after which the simplifications made in the model and its limitations are elucidated.

5.2.1 Modelled geometry

The main geometry of the stadium is described by three ellipses, one at ground level and two ellipses to describe the roof. Figure 5.5 shows the outer shape of the model. As stated in Section 5.1.3, the overall geometry of the stadium and its roof can be altered by changing parameters describing the two roof ellipses. These z-coordinates of these ellipses are modelled with a sine function. This means that the amplitude and number of extremes can be changed, together with a possible shift of the function. An example is shown in Figure 5.5. Other parameters which describe the shape of the model and are variables in the optimisation process are the height of the top ellipse relative to the other roof ellipse and the roof opening size. The parameter describing the roof opening size, is the main span. The minor span is 1.2 times smaller. Note that the description of the stands, field and rest of the interior of the stadium can be found in Section 5.2.3.

5. Case study

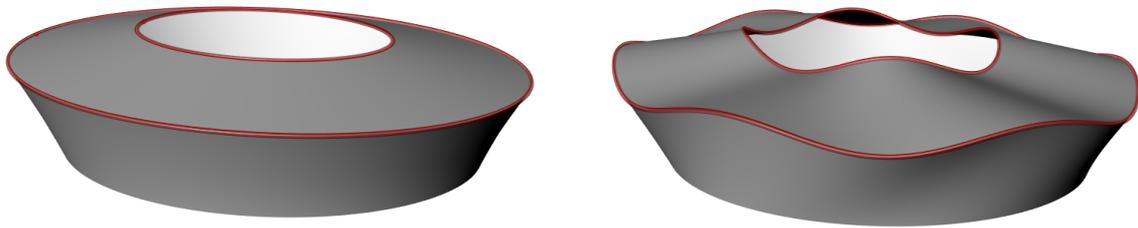


Figure 5.5: Outer shape of the model, with original Donbass Arena geometry (left) and altered geometry (right)

Simplifications in the modelled geometry

The main simplification which is made for describing the overall geometry of the stadium and its roof is the method of altering the shape of the roof. It was chosen to model both roof ellipses as sine functions, which limited the possible shapes of the roof. However, for more design freedom, a lot more parameters describing the geometry were needed for describing the shape. This had a large influence on the size of the design space and led to the choice for the sine functions.

5.2.2 Structural model

As stated in Section 5.1.1 the steel roof of the Donbass Arena consists of a network of steel elements. The roof is carried by 12 main cantilevering trusses, of which 4 are twice as big as the other 8. The roof structure is shown in Figure 5.2. The elements of the 12 cantilevering trusses are considerably bigger than the other elements. This, in combination with the connections between the cantilevers and the other elements, indicates that there is no compression ring at top of the roof. This is very important for the modelling of the structure, as a compression ring would lead to a different structural system.

The Karamba plug-in was used to model and analyse the roof structure in Grasshopper. As the roof model consisted 12.048 elements which needed to be optimised, it was chosen to reduce the roof model to the 12 cantilevering trusses. This will be further explained in Section 5.2.2. The main variables which describe the structural model are the earlier mentioned parameters describing the overall shape, the number of divisions in the trusses, the structural height of the trusses at both ends and the cross-sections of the elements. These cross-sections are driven by a separate cross-section optimiser, as explained in Section 3.2.2 and Section 4.2. The material of the elements is S275. The structural model of the roof is shown in Figure 5.6. Note that the blue elements are Karamba elements, the red elements are not checked.

The roof structure is checked for 4 load combinations. These load combinations are described in Table 5.3. The utilisation of the elements and the deformations of the structure are calculated by Karamba and checked by the given constraints (Section 5.1.4). It should be noted that Karamba outputs the utilisation and deformation values for the most unfavorable load combination.

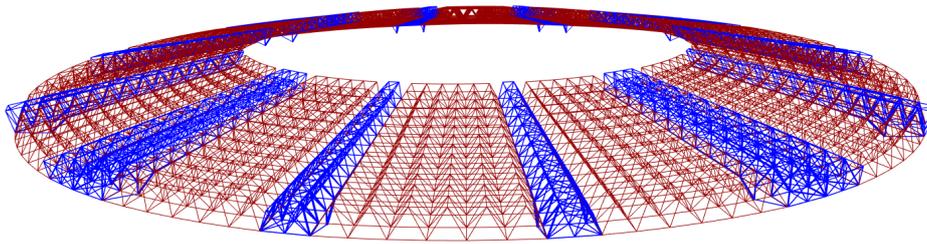


Figure 5.6: Structural model of the Donbass roof, with Karamba elements in blue

load combination 1	1.35	· total permanent load		
load combination 2	1.2	· total permanent load	+ 1.5	· snow load
load combination 3	1.2	· total permanent load	+ 1.5	· wind load (y)
load combination 4	1.2	· total permanent load	+ 1.5	· wind load (x)

Table 5.3: Load combinations applied on roof structure

Simplifications in the structural model

Several simplifications were made in the structural model, which need to be kept in mind when looking at the results. The simplifications are listed in this section. Section 5.2.4 will focus more on the possible consequences of these simplifications.

The first simplification made in the structural model, was already mentioned earlier. The original model consisted out of 12.048 structural elements, which proved to be difficult computationally. This is inconvenient for an optimisation process. The structural system of the Donbass Arena however allowed the structural model to be reduced to only the 12 main cantilevering trusses, which carry the remaining elements. This reduction decreased the number of elements to 4284, 35% of the original amount. The weight of the elements, along with the forces acting on them, were modelled as extra loading applied on the cantilevering trusses.

As stated, the structural model of the roof was checked for 4 load combinations. However, calculations for such a large structure would normally contain significantly more different load combinations. As employing all possible combinations was too labour intensive and computationally heavy, it was chosen to only use 4 of the principle load combinations. It was important to implement different load combinations to investigate the way Karamba dealt with this.

Table 5.3 shows that the loads are modelled in the Ultimate Limit State (ULS). However, the maximum allowable deflections of a structure apply to the Serviceability Limit State (SLS). As this is not possible to implement in Karamba in the conventional way, it was chosen to increase the given maximum allowable deflection. Testing showed that the maximum deflection in ULS is 1.325x as big as the maximum deflection in SLS in this model. To make a conservative assumption, the maximum allowable deflection is multiplied with 1.25.

5. Case study

Finally, it was chosen to focus the cross-section optimisation on circular hollow sections only. Karamba is only capable to optimise a group of elements for one kind of cross-section. With the earlier mentioned deviations with lateral torsional buckling (see Section 3.2.3) in mind, it was chosen to assign circular hollow sections to all elements. All of this has the consequence that the cross-section optimiser has a choice from 455 different sections instead of 6602 (all potential Karamba cross-sections).

Verification of the structural model

To check whether or not the structural model contains large errors, the normative cantilever was modelled in GSA as well. The first check was to compare the support reactions and normal forces in some elements. These are shown in Table 5.4 and Table 5.5. The checked elements and supports that are indicated in Figure 5.7. Note that only load combination 1 and 2 are compared.

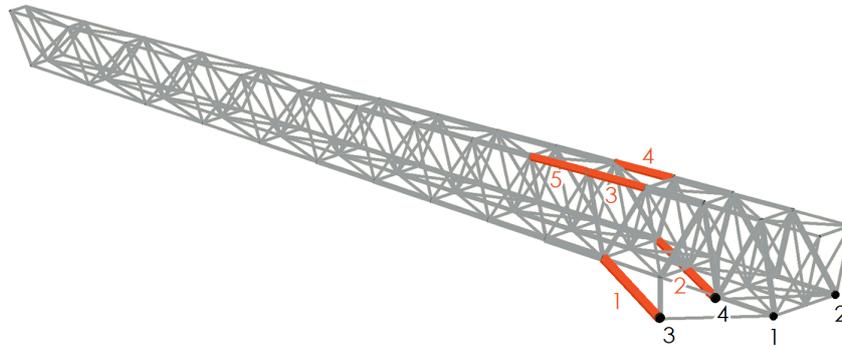


Figure 5.7: Indices of the checked normative cantilever

Element	Cross-section	Load combination	N in Karamba (kN)	N in GSA (kN)	Difference
1	CHS610x30	A1	-3733	-3712	0.6 %
		A2	-6210	-6189	0.3 %
2	CHS610x30	A1	-3821	-3800	0.5 %
		A2	-6358	-6340	0.3 %
3	CHS610x30	A1	4674	4654	0.4 %
		A2	7957	7941	0.2 %
4	CHS610x30	A1	4562	4539	0.5 %
		A2	7767	7747	0.3 %
5	CHS508x30	A1	3899	3887	0.3 %
		A2	6712	6707	0.1 %

Table 5.4: Comparison of the normal forces in the normative cantilever as determined by Karamba and GSA

		Fx (kN)		
Support	Load combination	Karamba	GSA	Difference
1	A1	666.9	662.9	0.6 %
	A2	1148.8	1145	0.3 %
2	A1	-2085.3	-2078	0.4 %
	A2	-3512.1	-3508	0.1 %
3	A1	562.9	569.4	-1.2 %
	A2	937.6	950.8	-1.4 %
4	A1	855.6	845.5	1.2 %
	A2	1425.7	1413	0.9 %

		Fy (kN)		
Support	Load combination	Karamba	GSA	Difference
1	A1	2917.3	2889	1.0 %
	A2	4868.7	4831	0.8 %
2	A1	2389.5	2379	0.4 %
	A2	3974.4	3968	0.2 %
3	A1	-2661	-2637	0.9 %
	A2	-4433.7	-4402	0.7 %
4	A1	-2645.8	-2631	0.6 %
	A2	-4409.5	-4396	0.3 %

		Fz (kN)		
Support	Load combination	Karamba	GSA	Difference
1	A1	-2901.8	-2884	0.6 %
	A2	-4922.3	-4902	0.4 %
2	A1	-2903.2	-2894	0.3 %
	A2	-4925.5	-4922	0.1 %
3	A1	4658.5	4627	0.7 %
	A2	7640.2	7608	0.4 %
4	A1	4595.3	4574	0.5 %
	A2	7531.3	7517	0.2 %

Table 5.5: Comparison of the support reactions of the normative cantilever as determined by Karamba and GSA

5. Case study

As can be seen in the comparisons, the differences between GSA and Karamba are negligibly small and it seems the Karamba structural model is set up accurately. However, the optimisation process is not guided by internal forces but by the utilisation of the elements and their displacements. Figure 5.8 shows the deviations of the utilisation of the elements between Karamba and GSA. All elements of the normative cantilever (268) were checked for both load combinations. What stands out, is that Karamba mainly overestimates the utilisation in comparison with GSA. Section 3.2.3 explains more about this. The mean deviation in utilisation is 3.38%. These trends are also observed in the benchmarking case shown in the mentioned section.

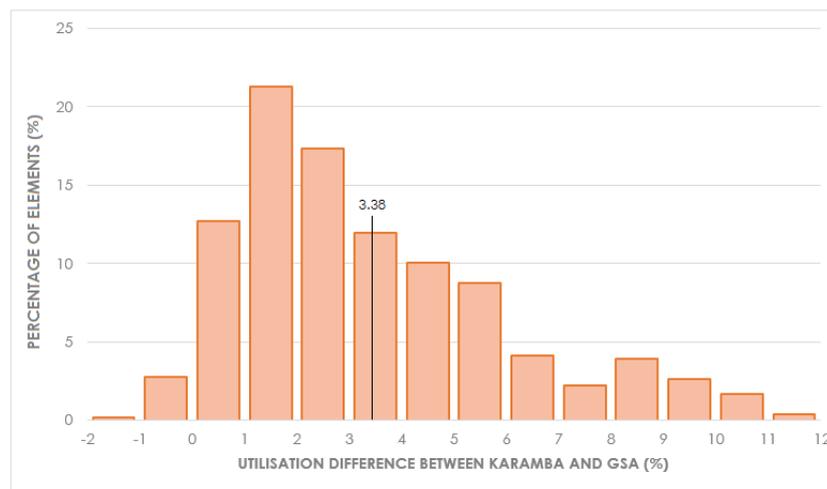


Figure 5.8: Histogram of the utilisation difference between Karamba and GSA for the normative cantilevering truss

The influence of small deviations in utilisation on the overall mass is only small, as only few elements would be affected so much that the cross-section would need to be altered. On top of that, utilisation is not for all cases governing. For example, if the constraint of displacement is removed in the baseline design and the maximum allowable utilisation is increased with 10%, the structural mass of the baseline design only decreases with about 1.8%. This deviation could increase when the span of the truss is enlarged or the number of elements increases.

Table 5.6 shows the deviations between Karamba and GSA in maximum displacement of the normative cantilevering truss for both load combinations. These deviations are slightly bigger than found for the utilisation, but also in determining the displacement of the truss, Karamba is more conservative. When the maximum allowable displacement is increased with 10%, the structural mass of the baseline design decreases with about 3.5% (from 3292.5 to 3177.4 kg tonnes). It should be noted that the deviation could be larger when the span of the trusses increases.

These margins should be kept in mind when interpreting the results, which means it could be assumed that the structural mass for the found solutions could be decreased with about 0 to 5%, depending on the number of elements in the trusses and the size of span. It was however chosen not to incorporate this in the post processing of the results, as it could lead to non-conservative assumptions.

Load combination	Max. displacement Karamba (m)	Max. displacement GSA (m)	Difference
A1	0.352	0.321	8.8 %
A2	0.603	0.558	7.5 %

Table 5.6: Maximum displacement of the normative cantilevering truss as found in Karamba and GSA

5.2.3 Acoustic model

The acoustic model of the Donbass Arena is built up from different elements. See Section 3.1 for a more elaborate explanation of the acoustic analysis. Figure 5.9 shows an overview of these different elements. The receiving surfaces (79) are shown in blue and represent the crowd and field, the listening positions at which acoustics are important. The black surface represents the opening between the roof and stand, through which sound has opportunity to escape the stadium bowl. This is modelled as a surface which absorbs all sound. The white points in the bowl are the source points and represent the crowd noise. As the crowd noise is mostly aimed at the pitch, the sources are not omni-directional but in the shape of a quarter of a sphere in direction of the pitch (see Figure 5.9) . Finally, the orange surfaces are the reflecting surfaces. Note that the roof and facade of the stadium are also reflecting surfaces, but are not shown in this figure.

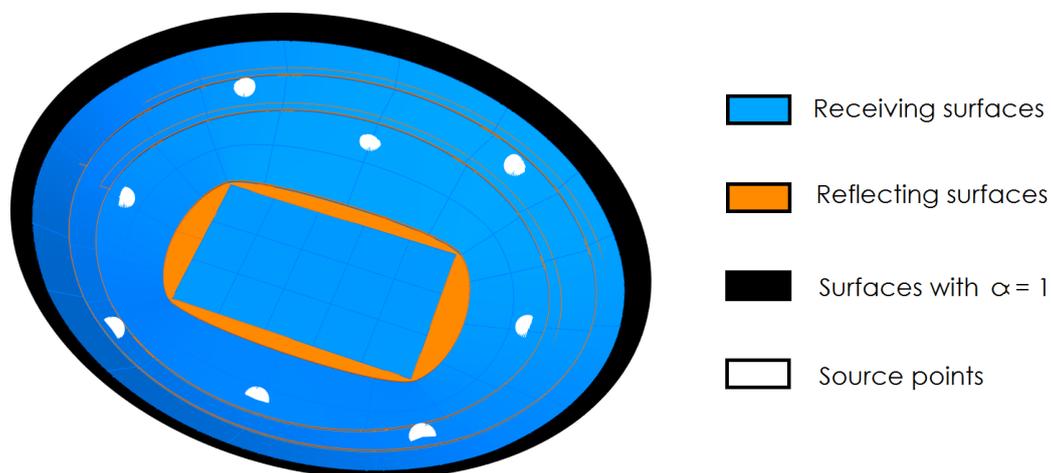


Figure 5.9: Overview of the elements in the acoustic model

To determine the sound pressure levels at the receivers, the acoustic model needs several more input (see Section 3.1). First of all, the absorption coefficient for the reflecting surfaces need to be provided . These are assumed as shown in Table 5.7. Note that a frequency of 1000 Hz was assumed, as most sound energy of human speech is around this frequency (Jack, 2013).

5. Case study

Glass facade	0.06
Steel roof (incl. elements)	0.12
Grass field	0.90
Concrete elements	0.05
Air absorption	0.012

Table 5.7: Absorption coefficients applied in the model

As stated, it was chosen that the sources radiate sound in the direction of a quarter sphere. This means that the directivity factor (Q) is 4. The number of rays emitted by all sources in this direction is about 600, thus around 5.000 in total. The maximum length of the rays is 343 meters. This is the distance that sound travels within one second. Early sound energy is mostly dependent on room shape while late sound energy is more dependent on the absorption (Echenagucia et al., 2014).

Simplifications in the acoustic model

To limit the amount of computation needed for assessing the acoustic performances, several simplifications of the model were needed. The first and main simplification is that of modelling the crowd. Ideally, the crowd would be modelled as 50.000 separate sound sources and receivers. It is however obvious this is not nearly possible. After several tests, it was chosen to model the crowd noise with 8 sources equally divided over the stands and the amount of receiving surfaces was set at 79. The difference with the 50.000 is considerable, which is at the expense of the level of detail in the sound mapping. As the sound pressure levels are determined per receiving surface, very localised abnormalities are not detected. The limited number of sources may cause peaks in sound pressure levels around these source, which in reality would be distributed over the entire stand. Also the 'optimal' roof shapes could be influenced by these limited amount of sources and receivers, which should be kept in mind when the results of the optimisation process are analysed.

It should also be noted, that the structural elements in the roof are also not taken into account when determining the reflections and ray paths, as this would lead to a too large amount of reflecting surfaces. The possibility of the sound getting 'trapped' in between these elements was modelled by increasing the absorption factor to the roof.

Verification of the acoustic model

The verification of the acoustic model is performed with the earlier mentioned software CATT Acoustic (CATT, 2015). It was chosen to compare three designs, acoustically assessed by both the custom tool (SPL-Estimator) and CATT. The three designs are the baseline design, a design with the maximum roof opening (opening of 105 meters) and a design with the minimal roof opening (35 meters). Note that for these cases, omnidirectional sound sources are applied as it is difficult to model the actual sound directions in CATT Acoustic. Table 5.8 shows the acoustic values which are found by SPL-Estimator and CATT for the described designs. To verify the

influence of the different acoustic parameters calculated by CATT on the acoustic objective, the sum of the acoustic deviation of the baseline design is taken equal to the deviation as calculated by SPL-Estimator. This was necessary as CATT Acoustic is not able to determine this value. The other acoustic deviations for CATT are determined in reference to this value.

	Roof opening (m)	μ (dB)	Difference	σ (dB)	Difference	E_{tot} (dB)	Difference
CATT	64	73.00	-	3.85	-	620.49*	-
SPL-Estimator	64	71.25	-2.5 %	4.44	13.3 %	620.49	-
CATT	35	73.11	-	3.80	-	611.50	-
SPL-Estimator	35	71.47	-2.3 %	4.34	12.4 %	603.27	-1.4 %
CATT	105	72.74	-	4.08	-	659.8	-
SPL-Estimator	105	70.25	-3.5 %	4.93	17.2 %	697.94	5.5 %

Table 5.8: Acoustic values for three described designs as determined by SPL-Estimator and CATT Acoustic

Similar to the findings in Section 3.1.3, this verification shows that SPL-Estimator mainly underestimates the mean value and overestimates the standard deviation of the sound pressure levels throughout the stadium. On top of that, the differences between performances of designs are also overestimated by SPL-Estimator. Consequently, the range of the acoustic objective is smaller than estimated by the custom tool. This means acoustically 'bad' solutions are actually better than estimated and acoustically 'good' solutions perform less than estimated, with an error up to 5.5%. However, SPL-Estimator is able to determine the trend in acoustic performances correctly and is thus able to rank the evaluated solutions from best to worst distribution of sound.

As the used sources in the case study are not omni-directional, the standard deviations of the case study results are smaller than in the verification case. For example, in the baseline design the average is 77.26 dB (instead of 71.25 dB) and the standard deviation is 2.41 dB (instead of 4.44 dB). This means the deviations between the estimation of the tool and the acoustic response is smaller than in the verification. The results should however be interpreted with the earlier stated errors in mind.

5.2.4 Limitations and inaccuracy

The earlier mentioned simplifications made to be able to model the Donbass Arena lead to limitations in terms of design freedom and inaccuracy in the assessment of the performances of the potential solutions. Consequently, the simplifications may influence the results and must be kept in mind during the analysis of those results. Next to the inaccuracy caused by the way the model is set up, it also defines the design space. Therefore, the optimal results that are found during the optimisation are optimal for the described optimisation problem and model.

5. Case study

If the model was to be set up differently, other optimal solutions are found. In case a solution is referred to as 'optimal' in the following chapters, the solution is only optimal within the given criteria (defined optimisation problem) and compared to the other solutions that are found.

The main limitations of the model are listed below.

- The variables are limited to the roof, the overall shape of the stadium and stands are fixed.
- The geometry of the roof is limited by the description of two sine functions, a roof opening and a height difference.
- The location of the main cantilevering trusses is fixed.
- The amount of divisions in the cantilevering trusses are equal for all trusses. This also applies for the structural height of the trusses.
- Only circular hollow sections are evaluated as structural elements.
- Material properties are not variable during the optimisation process; all materials are chosen beforehand
- Sound paths are calculated for only one second and one frequency.

In terms of inaccuracy due to simplifications in the model, the following deficiencies are to be kept in mind when the results are assessed. Some deficiencies may be mentioned earlier in this chapter already. Inaccuracy due to the used analysis software are indicated in earlier sections.

- As the structural part of the roof is limited to the cantilevering trusses, the influence of the geometry of the remainder of the roof is not taken into account. A highly curved roof might have a higher weight per square meter than a straight roof. The weight per square meter (for the trusses) is therefore estimated 25% higher than calculated for the baseline design. Consequently, the mass of relatively straight roofs are overestimated.
- Local deviations of the roof geometry are not taken into account in determining the snow and wind loads. Highly curved roofs might endure higher loads due to snow accumulation of local wind loads.
- Due to the limited amount of sources used for the acoustic model, solutions might be focused around these source points.
- Due to the limited amount of receiving surfaces, local effects and acoustic deviations might be missed.

5.3 Baseline design

In this section, the baseline design is illustrated. The variables describing the baseline design are shown in Section 5.1.3. It was tried to model the baseline design as close to the actual Donbass Arena as possible. The performances of the baseline design are shown in Table 5.9. As can be seen, the structural mass is lower than the actual roof structure. This is because roof structure was modelled with circular hollow sections, while the cantilevering trusses in reality are I-profiles. These sections are however not ideal for modelling in Karamba (see Section 3.2.3). Figure 5.10 shows the modelled baseline design.

Structural mass (kg tonnes)	3292.5
Acoustic 'error' (dB)	551.2
Crowd covered by roof (%)	97.4
Lost sound through roof opening (%)	15.5

Table 5.9: Performances of the baseline design

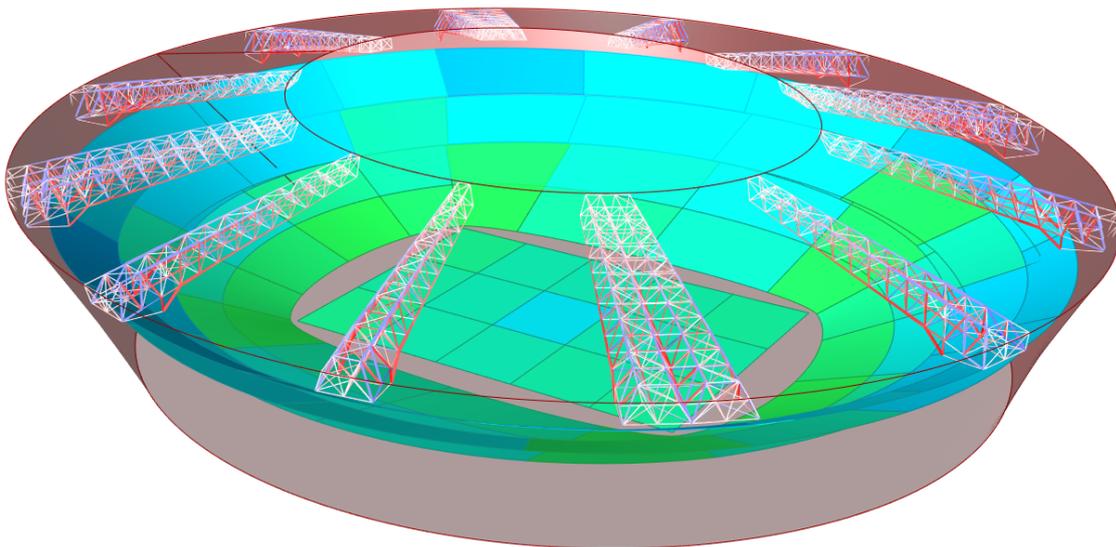


Figure 5.10: Model of the baseline design

6

Optimisation results

In this chapter, the optimisation results of the described case study (see chapter 5) are presented. The optimisation process ran on a computer with an Intel(R) Xeon(R) CPU E5-1620 0 @ 3.60GHz 3.60Ghz processor and 32.9 GB of installed memory (RAM) for about 32 hours, divided over 5 sessions. It was able to evaluate 1.116 solutions, which means each solution took about 100 seconds to assess. 808 (72.4%) of these solutions were feasible (satisfied the design constraints) of which 33 were 'Pareto optimal' solutions (4.1% of the feasible designs). Note that these solutions are 'optimal' within the predefined optimisation problem and criteria.

6.1 Optimal solutions

The solutions (both feasible and infeasible) are shown in Figure 6.1, which is produced by the custom post processing tool (PPOR). Compared with the baseline design, the best performing feasible design in terms of acoustics is improved by 22.8%. Looking at the other extreme, the mass of the best 'structural' solution is 58.3% lighter than the baseline design. To verify the results of PPOR, Figure 6.2 shows the Pareto front generated by both Octopus and PPOR. As can be seen in the figure, the found Pareto fronts are identical. An example of how the results are presented by PPOR is given in Appendix C.

The shape of the Pareto front (Figure 6.1) is remarkable. Solutions in some parts of the front are very close to each other, whilst in other areas Pareto solution are scarce. Furthermore, the Pareto front does not seem to be a smooth line. Looking at the infeasible designs (in gray), it seems like there are some areas in the design space where solutions are harder to be found. As only a limited amount of designs are evaluated, the algorithm has not been able to find a feasible design in these areas. Chapter 7, will elaborate further on this.

6. Optimisation results

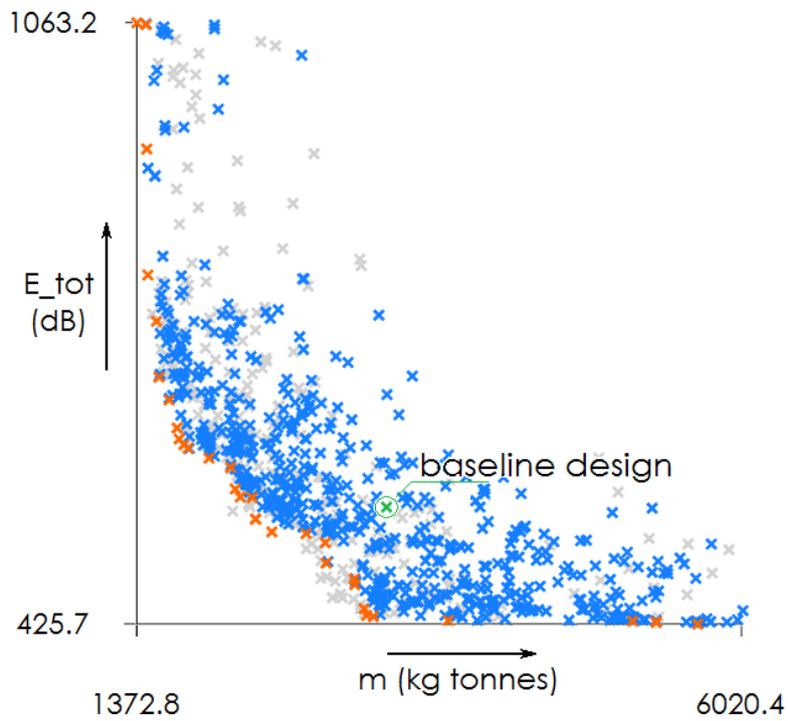


Figure 6.1: Scatter plot of the feasible case study results (Pareto front in orange, unfeasible solutions in gray and baseline design in green)

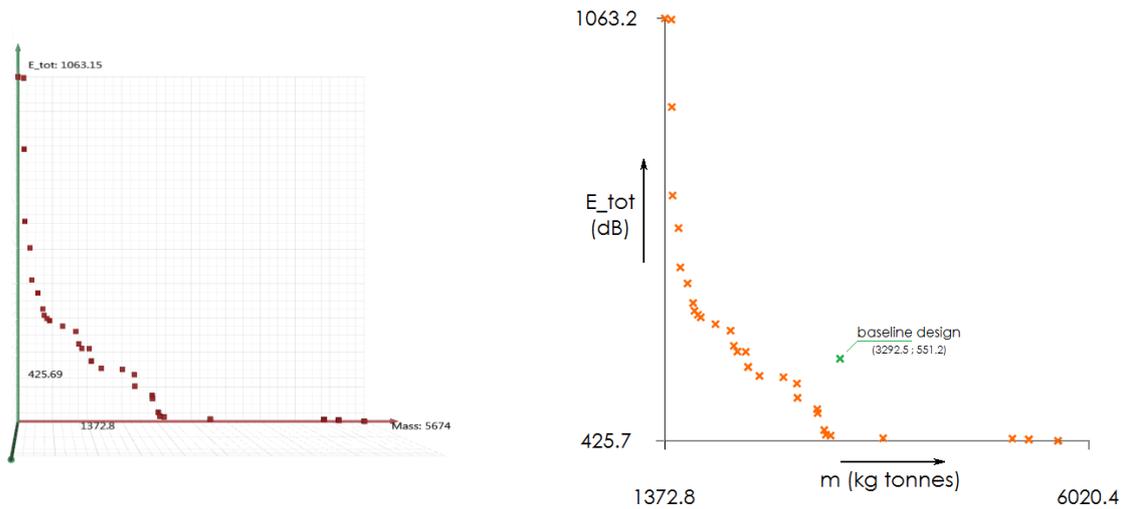


Figure 6.2: Trade off between the acoustic and structural objective in Octopus (left) and PPOR (right)

Table 6.1 shows the performances of the Pareto optimal solutions. Also the differences in performance with the baseline design are shown. What stands out, is that solutions at both outer edges of the Pareto front perform only slightly better for one discipline than the other Pareto solutions, but perform significantly worse for the other discipline compared with the rest of the solutions. Looking at the results, it appears this is consequence of the too small amount of evaluated solutions during the optimisation process. In Chapter 7, this is investigated further.

	m (tonnes kg)	E_{tot} (dB)	Structural improvement (%)	Acoustic improvement (%)
1.	1372.8	1063.2	58.3	-92.9
2.	1443.8	1061.2	56.1	-92.5
3.	1449.5	929.4	56.0	-68.6
4.	1457.5	795.8	55.7	-44.4
5.	1523.4	746.7	53.7	-35.5
6.	1543.1	687.3	53.1	-24.7
7.	1621.0	663.2	50.8	-20.3
8.	1681.6	633.7	48.9	-15.0
9.	1697.3	622.0	48.4	-12.8
10.	1734.0	615.8	47.3	-11.7
11.	1765.4	612.0	46.4	-11.0
12.	1926.8	601.7	41.5	-9.2
13.	2091.9	591.9	36.5	-7.4
14.	2127.7	569.0	35.4	-3.2
15.	2166.3	560.4	34.2	-1.7
16.	2258.0	560.0	31.4	-1.6
17.	2283.4	537.3	30.6	2.5
18.	2284.4	536.8	30.6	2.6
19.	2408.0	523.6	26.9	5.0
20.	2670.6	521.7	18.9	5.4
21.	2818.3	512.0	14.4	7.1
22.	2824.6	490.6	14.2	11.0
23.	3040.3	473.5	7.7	14.1
24.	3044.4	467.5	7.5	15.2
25.	3116.3	442.0	5.4	19.8
26.	3135.1	434.8	4.8	21.1
27.	3187.1	433.6	3.2	21.3
28.	3761.6	429.6	-14.2	22.1
29.	5174.8	429.0	-57.2	22.2
30.	5175.2	428.9	-57.2	22.2
31.	5354.5	428.1	-62.6	22.3
32.	5355.4	427.4	-62.7	22.5
33.	5674.0	425.7	-72.3	22.8

Table 6.1: Performance values of Pareto front solutions

As stated earlier, the results can also be shown in a parallel plot. In that way, the values of the variables defining a solution are also shown. Figure 6.3 shows the parallel plot, with all feasible solutions. Orange poly lines represent Pareto optimal solutions. Figure 6.4 shows the

6. Optimisation results

same parallel plot, but with solely the Pareto solutions. Note that the plot is more clear on a computer, as can be browsed through the different solutions.

From these plots, it stands out that all Pareto optimal solutions have a smaller mean height difference between the roof edges than the baseline design. Most optimal solutions have a difference between -5 and 5 meter, while the original design has a difference of 12 meter. Figure 6.4 again clearly indicates that for some optimal solutions, only a small increase in mass could lead to significant improvements acoustically. As the results are difficult to assess 'on paper' in these plots, some of the results will be assessed and validated with scatter plots showing the trade-off between certain variables and objectives in Chapter 7.

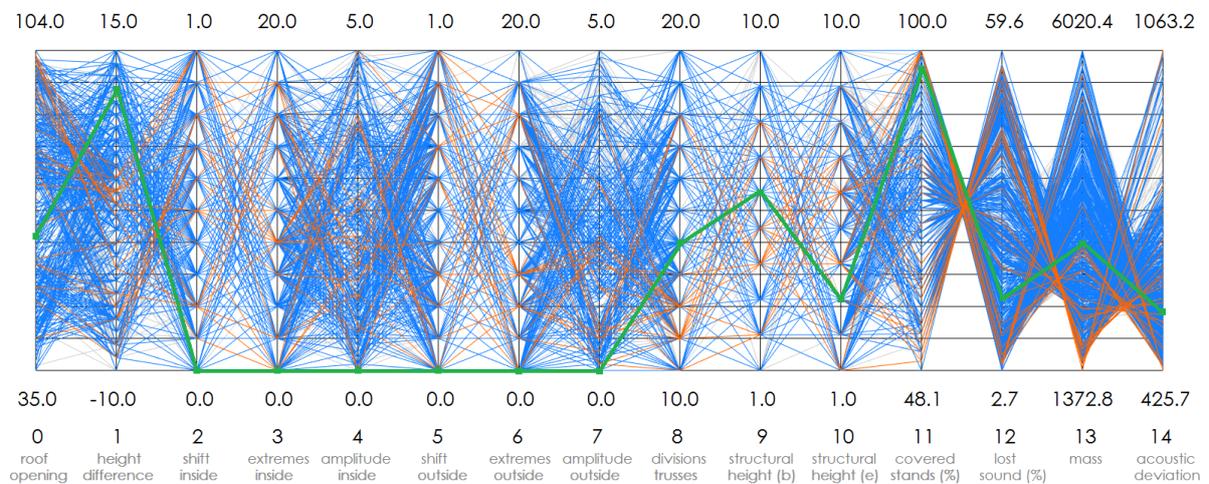


Figure 6.3: Parallel plot of feasible solutions (Pareto solutions in orange, baseline in green)

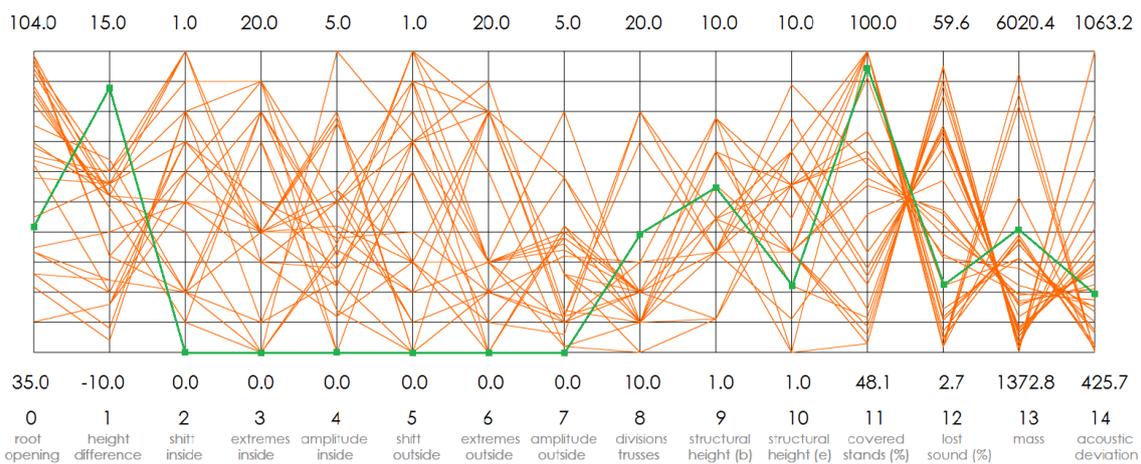


Figure 6.4: Parallel plot of Pareto optimal solutions (Baseline design in green)

6.2 Processing the results

To illustrate the possibilities of post processing with PPOR, this section will show a possible strategy to help find interesting solutions. As can be seen in the previous section, not all Pareto optimal solutions are equally good when the acoustic and structural performances are averaged. Moreover, as the performance of a building is not always the most important deciding factor (see Section 2.4), also dominated solutions could still be interesting design solutions. Note that this strategy is only an example; the post process strategy is largely dependent on a users intention.

The chosen strategy to exemplify the potential benefits of PPOR, is to filter out all solutions which perform worse than the baseline design in at least one discipline. This means that only designs remain with a better acoustic and structural performance, less sound loss and more stands covered by the roof. The 72 remaining designs (9% of all found feasible solutions) are shown in Figure 6.5 and Figure 6.6. Of these 72 designs, 6 solutions are Pareto optimal.

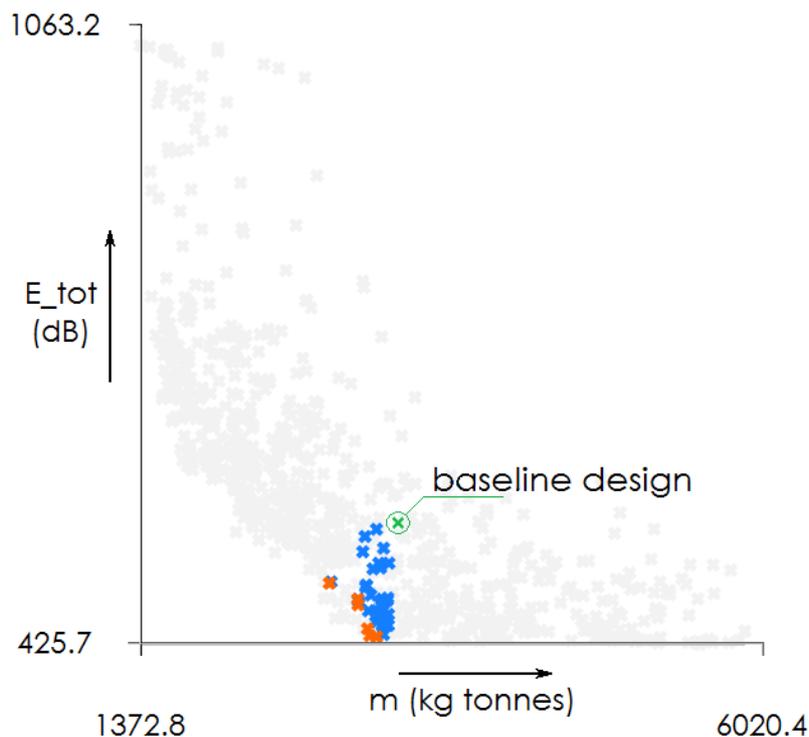


Figure 6.5: Scatter plot with solutions outperforming the baseline design (baseline indicated in green, Pareto optimal solutions in orange)

Looking at the parallel plot, some aspects stand out. The designs of the 6 remaining Pareto optimal solutions as they are shown by PPOR are presented in Figure 6.7. The designs are ordered from best structural solution to best acoustic solution (and worst structural solution). Note that these designs are solution 22 to solution 27 in Table 6.1. There are some more solutions outperforming the baseline design in both objectives, but these solutions either lose more sound or cover less stands than the baseline design.

6. Optimisation results

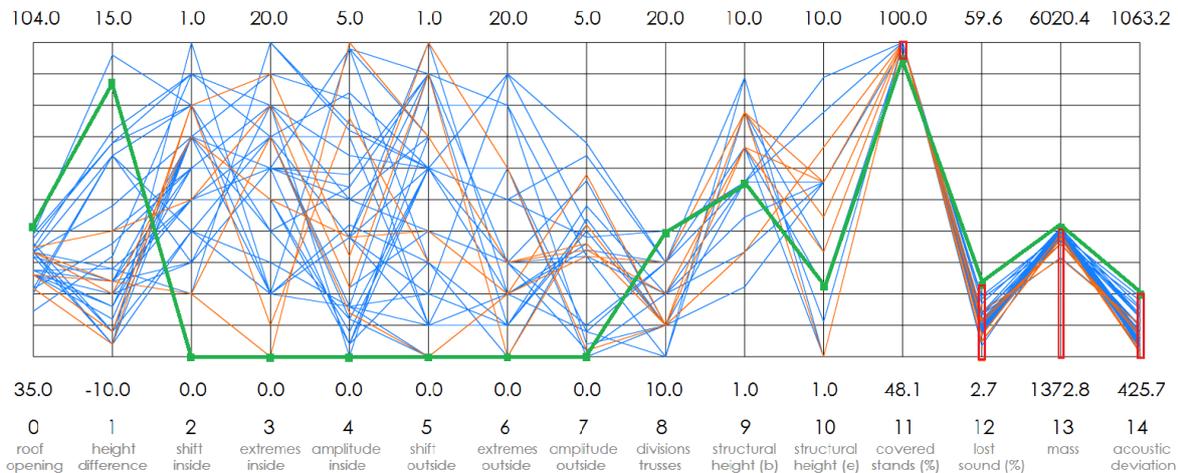


Figure 6.6: Parallel plot solutions outperforming the baseline design (baseline indicated in green, Pareto optimal solutions in orange)

- All roof openings all smaller than the roof opening of the baseline design, as at least the same percentage of the stands must be covered by the roof.
- Almost all solutions have a smaller mean height difference between the roof edges than the baseline design has. The optimal solutions all have a height difference smaller than 0.
- The amplitude of the inside roof edge ellipse tends to be bigger than the amplitude of the outside ellipse.
- Although all spans are equal or larger than the baseline design span, all solutions have a smaller amount of divisions in the cantilevering trusses than the baseline design has.
- Most solutions have a larger structural height at both sides of the truss than the baseline design has.

This example clearly shows that this multidisciplinary design optimisation does not only lead to better performing (optimised) solutions, but also provides more insight in the influence of different variables on the building performance. This potentially could lead to more informed decision making. In this post processing strategy, focus was given on the performances for the different disciplines. However, if this would lead to designs that are undesired for any reason (e.g. aesthetics), filters could also be applied on the design variables instead of performance indicators. In that way, the user is able to search for solutions with a certain characteristic instead of performance.

In the next chapter, the results will be analysed in more detail to validate them.

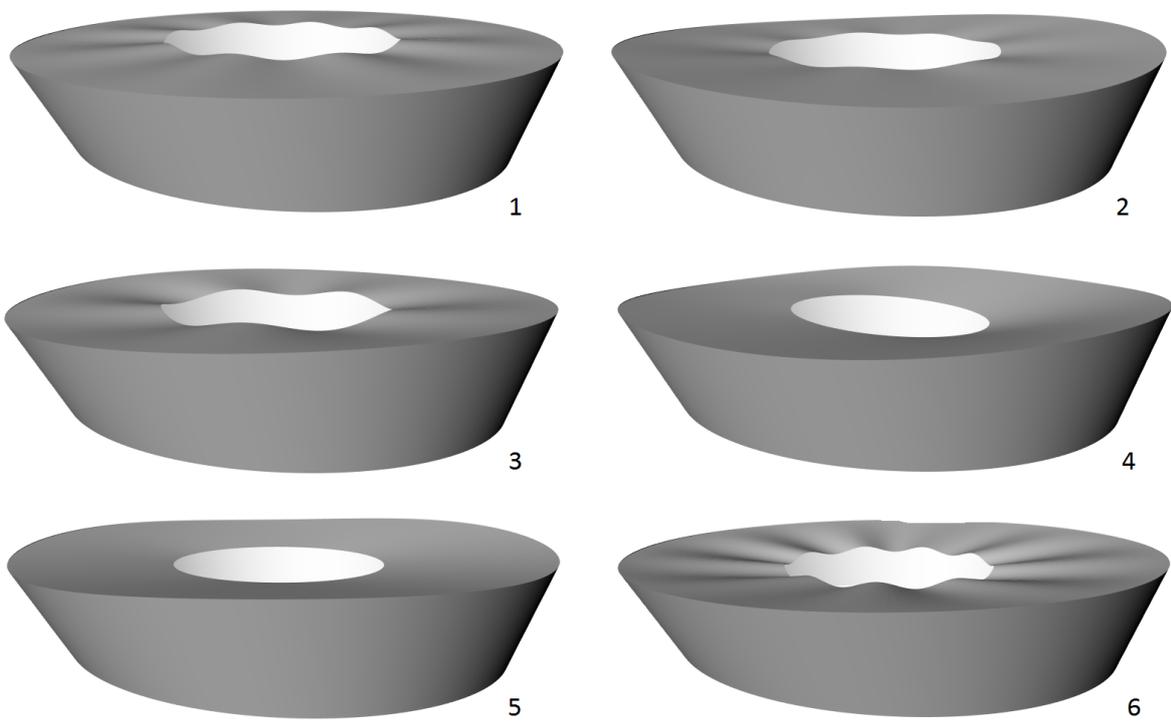


Figure 6.7: Designs of the Pareto optimal solutions outperforming the baseline design for all aspects

7

Validation

In this chapter, the optimisation results are validated. As the analysis methods (Chapter 3), optimisation approach, workflow, post processing tool PPOR (Chapter 4) and the model set up (Chapter 5) are investigated and verified before the optimisation was conducted this chapter will mainly focus on qualitatively investigating the trends detected in the results. In that way, it is examined if the model was set up correctly and if the findings correspond with known theory.

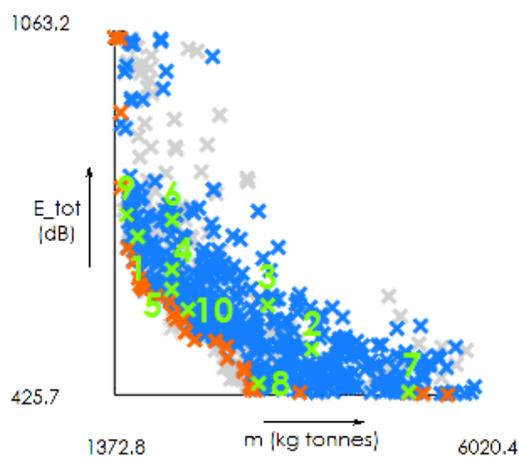


Figure 7.1: Scatter plot with the 10 validated results (in green)

Firstly, it is checked if the tool that was to assess and investigate the optimisation results is working as it should. To validate the used post processing tool PPOR for this case study and to be

7. Validation

sure that the results that are entered are correct, the variables of 10 random solutions are reinstated in the original model of the Donbass Arena. These 10 random solutions are shown in a scatter plot in Figure 7.1. The model determines the structural mass of roof and the acoustic deviation again and these recalculated values are compared with the values as presented by PPOR. As can be seen in Table 7.1, the recalculated structural mass and acoustic deviation perfectly correspond with the values taken from PPOR. It is therefor assumed that the results of the other solutions, as shown in PPOR, are also correct. The geometry, a cantilevering truss and the acoustic result of each of these 10 designs are shown in Figure 7.7, 7.8 and 7.9.

By investigating some scatter plots of different variables, the influences of these variables are further investigated and notable results are verified. The first variable that is investigated is the size of the roof opening. Figure 7.2 shows the trade-offs between the roof opening, the structural mass and acoustic deviation. Note that the Pareto solutions are shown in orange and the baseline design in green.

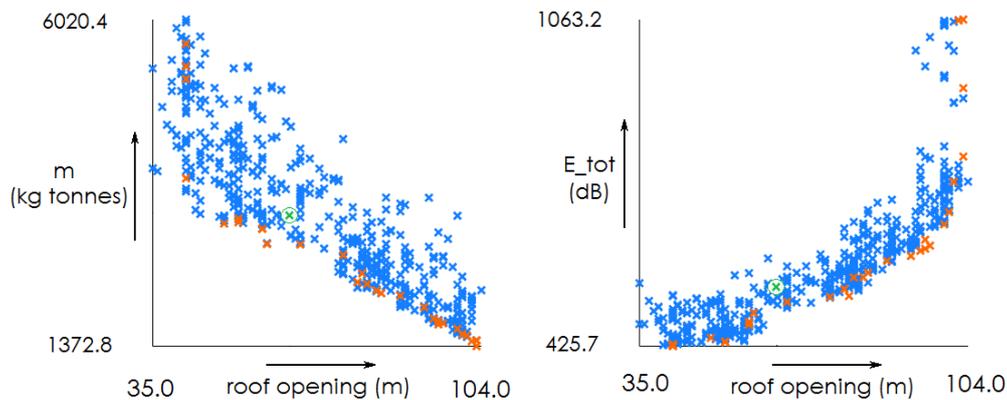


Figure 7.2: Scatter plot showing trade-offs between the roof opening, structural mass (left) and acoustic deviation (right)

A clear trend can be found in both plots. The bigger the size of the roof opening, the lower the mass of the structure is. This can be reasoned with the fact that the larger the roof opening is, the smaller the span is and thus less material is needed. On top of that, the internal forces increase with a larger span (Equation 7.1). Consequently, larger cross-sections are needed which also leads to mass increase. However, the acoustic deviation is contrasting with the structural mass. The larger the roof opening, the larger the structural deviation. This is consequence of the amount of sound that is lost through the opening.

$$M = \frac{1}{2} \cdot q \cdot l^2$$

with :

$$\begin{aligned} M &= \text{bending moment} \\ q &= \text{distributed load on cantilever} \\ l &= \text{span of cantilever} \end{aligned} \tag{7.1}$$

Figure 7.3 presents the trade-offs between the difference in height at the two roof edges, the structural mass and acoustic deviation. Note that the Pareto solutions are shown in orange and the baseline design in green. It shows that the roof height difference only has very little influence on the structural mass. The structurally best performing solutions all have a roof height difference around 0. However, note that roofs with small slopes could prove to be infeasible due to other restrictions (such as restrictions consequence to accumulating water). When the absolute difference increases, the solutions are becoming slightly heavier. This is consequence of the way the model is set up. As the span is defined horizontally (by the roof opening variable), the lengths of the cantilevers increases slightly when the absolute height difference increases (see Equation 7.1). However, the trade-offs also shows that all Pareto optimal solutions all fall in the range of -10 to 6. This is consequence of the influence of the acoustic deviation. A smaller volume of the stadium leads to acoustically better performing solutions, as the ray paths decrease in length (Owens, 2014). The acoustic trade-off also shows this trend (Figure 7.3, at the right). Noteworthy is that the chosen solution in the baseline design, is far outside the optimal range.

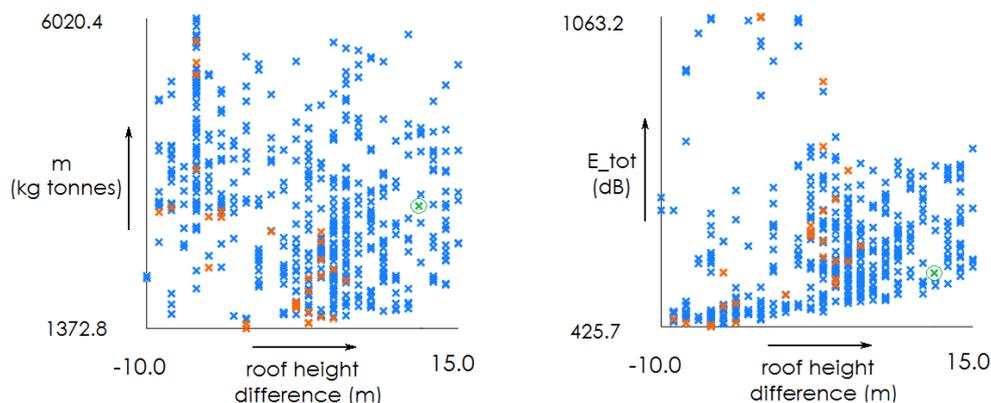


Figure 7.3: Scatter plot showing trade-offs between the roof height difference, structural mass (left) and acoustic deviation (right)

Next, the trade-offs between the number of divisions in the trusses and the structural mass and acoustic deviation are investigated (see Figure 7.4). What stands out the most, is that the Pareto optimal solutions are again mainly in the lower range of the variable, with the number of divisions used in the baseline design just outside this range. For this variable, this trend is caused by the structural objective (as it has no influence on the acoustic objective, see Section 5.1.3). Less divisions in the trusses lead to less elements, which however need bigger cross-sections. As expected, solutions at the outer edges are more scarce as these values more often lead to infeasible designs.

The last variable that is evaluated in this manner, is the structural height at the beginning of the cantilevering trusses (at the supports). The scatter plots are shown in Figure 7.5. Most solutions are concentrated in the middle of the variable range, as a (too) small structural height leads to higher stress and a too high structural height could lead to buckling problems. A low structural height shows to be slightly beneficial for the acoustic deviation, as it would enable the roof to be closer above the stands.

7. Validation

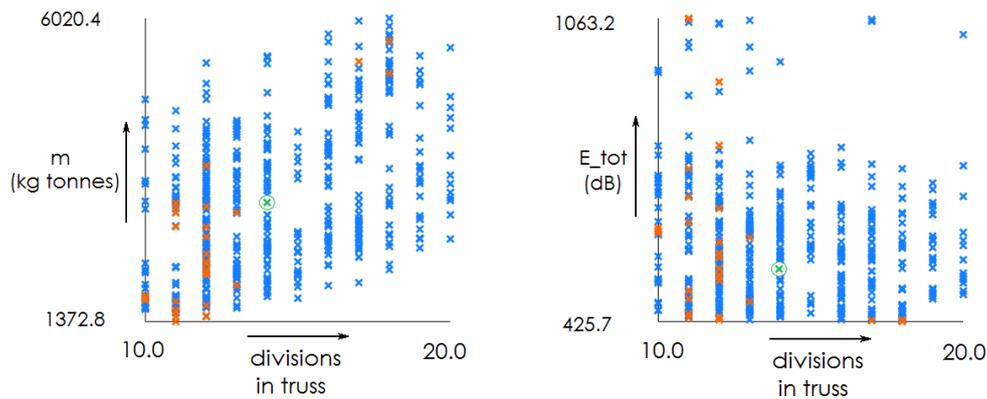


Figure 7.4: Scatter plot showing trade-offs between the number of divisions in the trusses, structural mass (left) and acoustic deviation (right)

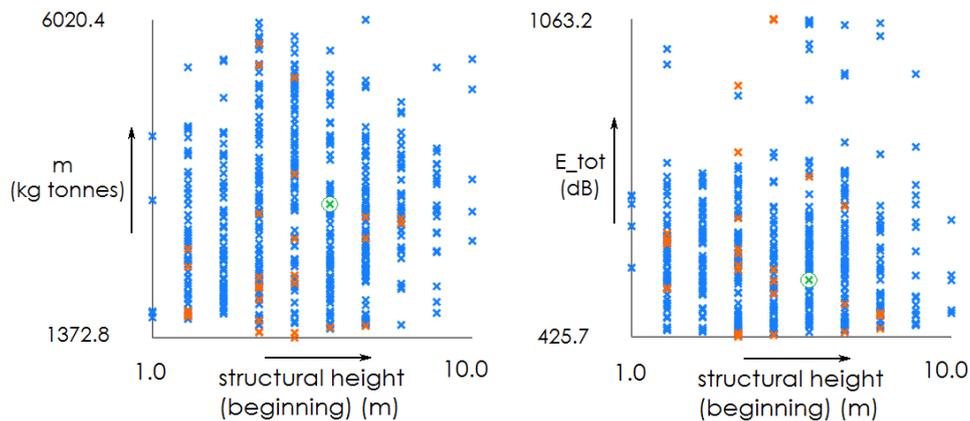


Figure 7.5: Scatter plot showing trade-offs between the structural height at the beginning of the cantilevering trusses, structural mass (left) and acoustic deviation (right)

The considered variables influence the objective functions mostly as could be expected. The scatter plots of the remaining variables do not show remarkable results. However, to investigate the influence of the roof geometry with respect to the shape of the edges, the scatter plots are not clear. As the geometry is described by 6 different variables, it is difficult to assess the results by looking at the separate scatter plots.

However, it is important to check the influence of the chosen location of the limited amount of sources, as the geometry of the roof should not be defined by this simplification. Figure 7.6 shows the geometry of the 6 Pareto optimal designs found during the post processing of the results (see Section 6.2) together with the locations of the 8 sources and the 12 cantilevering trusses. The sine shapes are represented by alternating blue (minimums) and red (maximums). Investigating these 6 designs, there is no real trend found around the 8 source points. As these results belong to some of the acoustically best found solutions, it is assumed that the found results are not (yet) affected by this made simplification. If the optimisation process would

have run longer and had evaluated more solutions, this could however deteriorate.

Finally, an investigation is done to what extent the optimisation algorithm was able to estimate the Pareto front. Especially when solutions at the outer edges and at the indentations of the estimated Pareto front are examined, it appears the solutions are still easily improvable. Looking at Pareto front solutions 28 and 29 for example (see Table 6.1), the difference in acoustic deviation is only 0.1%. However, the difference in structural mass is 43%. When the variables are examined, the designs are very similar, except for the number of divisions in the cantilevering trusses. This number is 18 for solution 29 and only 12 for solution 28. When the amount is decreased to 12 divisions for solution 29, the structural mass is decreased to 3761.9 kg tonnes (instead of 5174.8 kg tonnes). Also solutions at the indentations are still improvable by hand. For example, lowering the roof height difference from 4 to -2 for solution 21 (Table 6.1) leads to further improvement of both the structural and acoustic objective (512.0 dB to 498.4 dB and 2818.3 kg tonnes to 2785.7 kg). This shows that not enough solutions are evaluated by the algorithm at this point. The insights for these improvements are however discovered with the help of the optimisation results and PPOR.

With these arguments and verifications, it is assumed that the results of the optimisation process are valid. It is however important to keep in mind that these results can have deviations, due to the applied analysis methods and modelling (see Chapter 3 and Chapter 5). Furthermore, it should be noted that the optimisation process was not able to evaluate sufficient solutions to assume the estimated Pareto optimal solutions are actually 'optimal'.

7. Validation

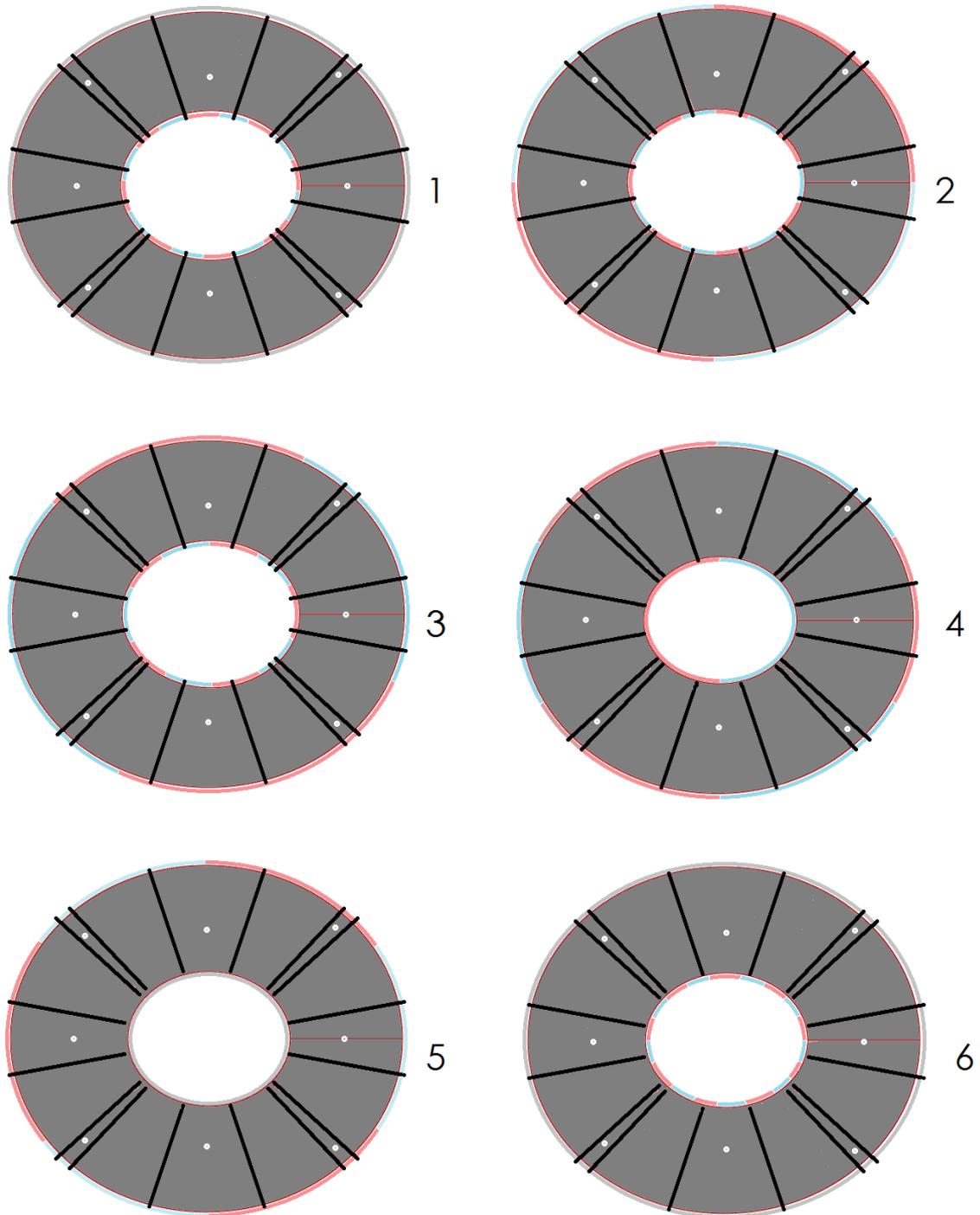


Figure 7.6: Topview of the evaluated solutions, with the 8 source points (white), cantilevering trusses (black) and the sine shapes (blue and red)

	1	2	3	4	5	
Variable 0	100	64	77	93	89	..
Variable 1	11	-2	7	5	10	..
Variable 2	0.3	0.2	0.2	0.5	0.4	..
Variable 3	8	12	16	10	16	..
Variable 4	0.5	4.5	2.5	0.4	1.1	..
Variable 5	0.8	0.9	0.0	0.9	0.2	..
Variable 6	6	10	14	0	2	..
Variable 7	3.9	1.6	3.4	3.4	1.9	..
Variable 8	11	20	20	13	13	..
Variable 9	1	9	7	8	3	..
Variable 10	7	4	2	10	7	..
E_tot (dB)	707.8	506.8	585.4	649.3	613.7	..
mass (kg tonnes)	1670	3916	3350	2114	2130	..
Pareto ?	FALSE	FALSE	FALSE	FALSE	FALSE	..
Recalculated E_tot (dB)	707.8	506.8	585.4	649.3	613.7	..
Recalculated mass (kg tonnes)	1670	3916	3350	2114	22130	..
Difference E_tot (%)	0	0	0	0	0	..
Difference mass (%)	0	0	0	0	0	..
		6	7	8	9	10
Variable 0	..	100	42	53	101	84
Variable 1	..	13	-6	-7	5	6
Variable 2	..	1.0	0.6	0.7	0.9	0.2
Variable 3	..	0	8	18	18	4
Variable 4	..	0.6	1.1	4.1	2.5	1.2
Variable 5	..	0.8	0.9	0.1	0.8	0.1
Variable 6	..	16	16	12	6	10
Variable 7	..	1.3	0.6	0.0	4.0	0.0
Variable 8	..	15	18	11	11	13
Variable 9	..	5	5	4	6	4
Variable 10	..	8	7	7	4	2
E_tot (dB)	..	737.7	431.0	445.7	746.7	577.7
mass (kg tonnes)	..	2115	5170	3233	1523	2321
Pareto ?	..	FALSE	FALSE	FALSE	TRUE	FALSE
Reinstated E_tot (dB)	..	737.7	431.0	445.7	746.7	577.7
Reinstated mass (kg tonnes)	..	2115	5170	3233	1523	2321
Difference E_tot (%)	..	0	0	0	0	0
Difference mass (%)	..	0	0	0	0	0

Table 7.1: Validation of 10 random solutions, manually reinstated in Grasshopper

7. Validation

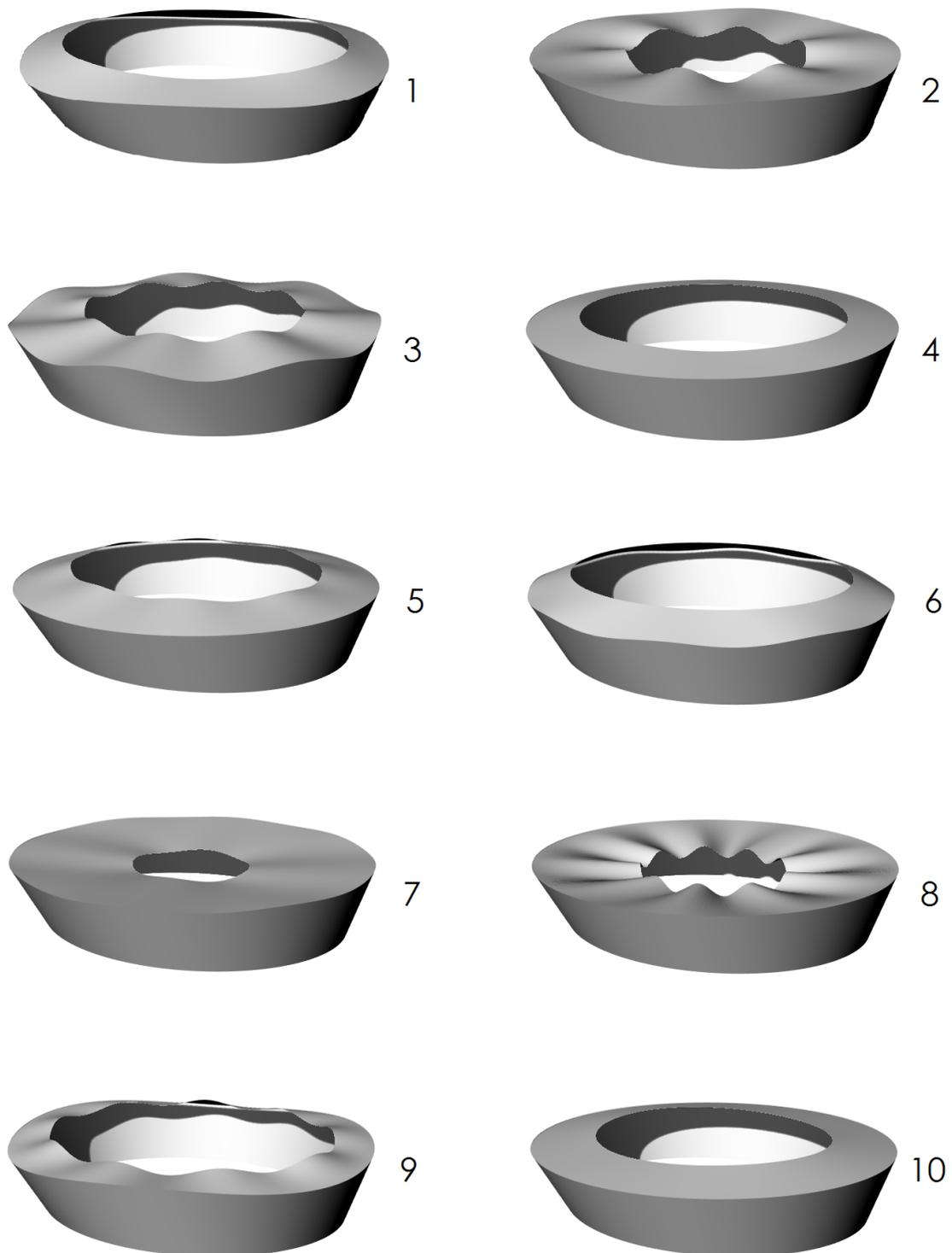


Figure 7.7: Geometry of each of the 10 validated results

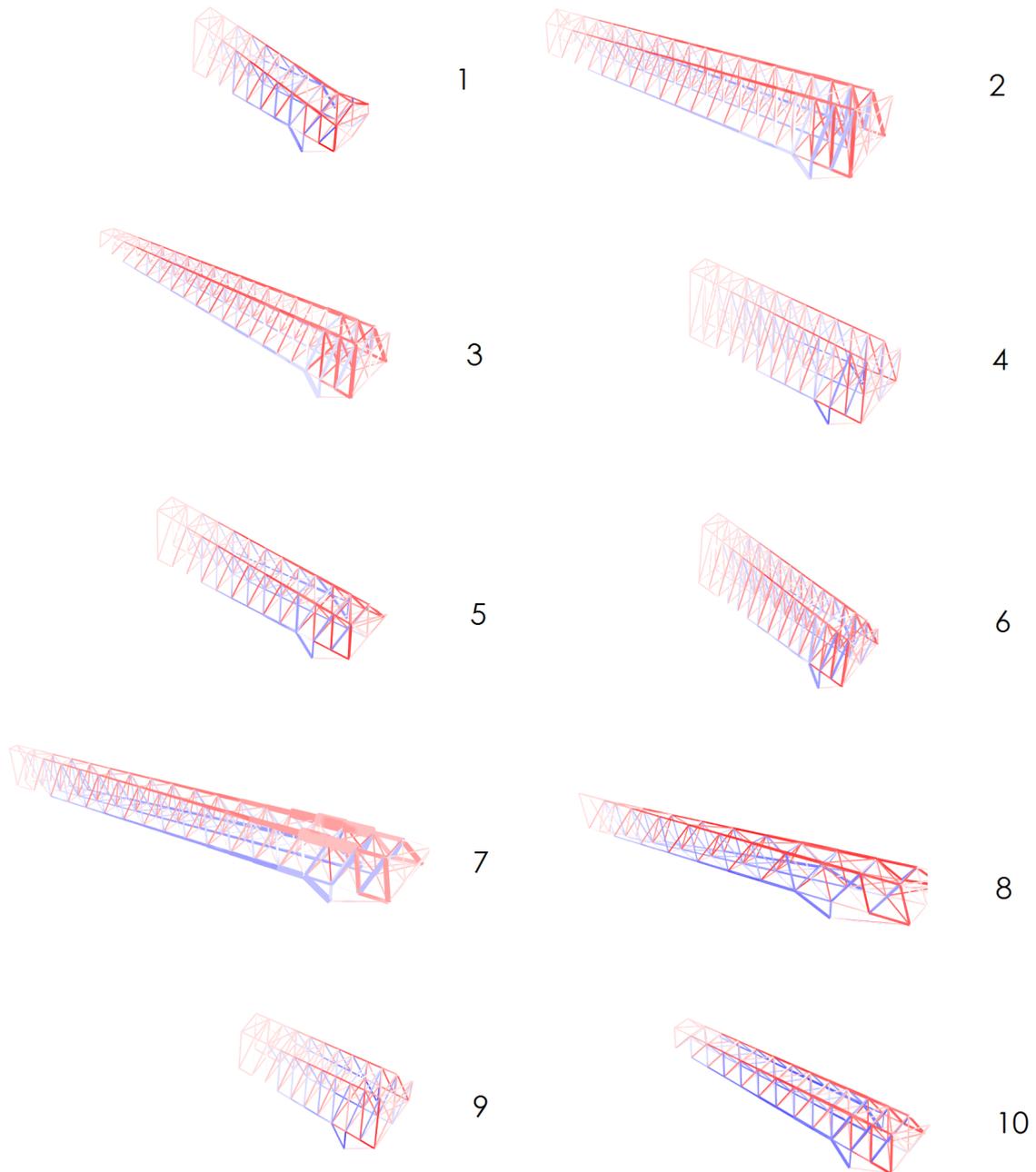


Figure 7.8: A cantilevering truss of each of the 10 validated results

7. Validation

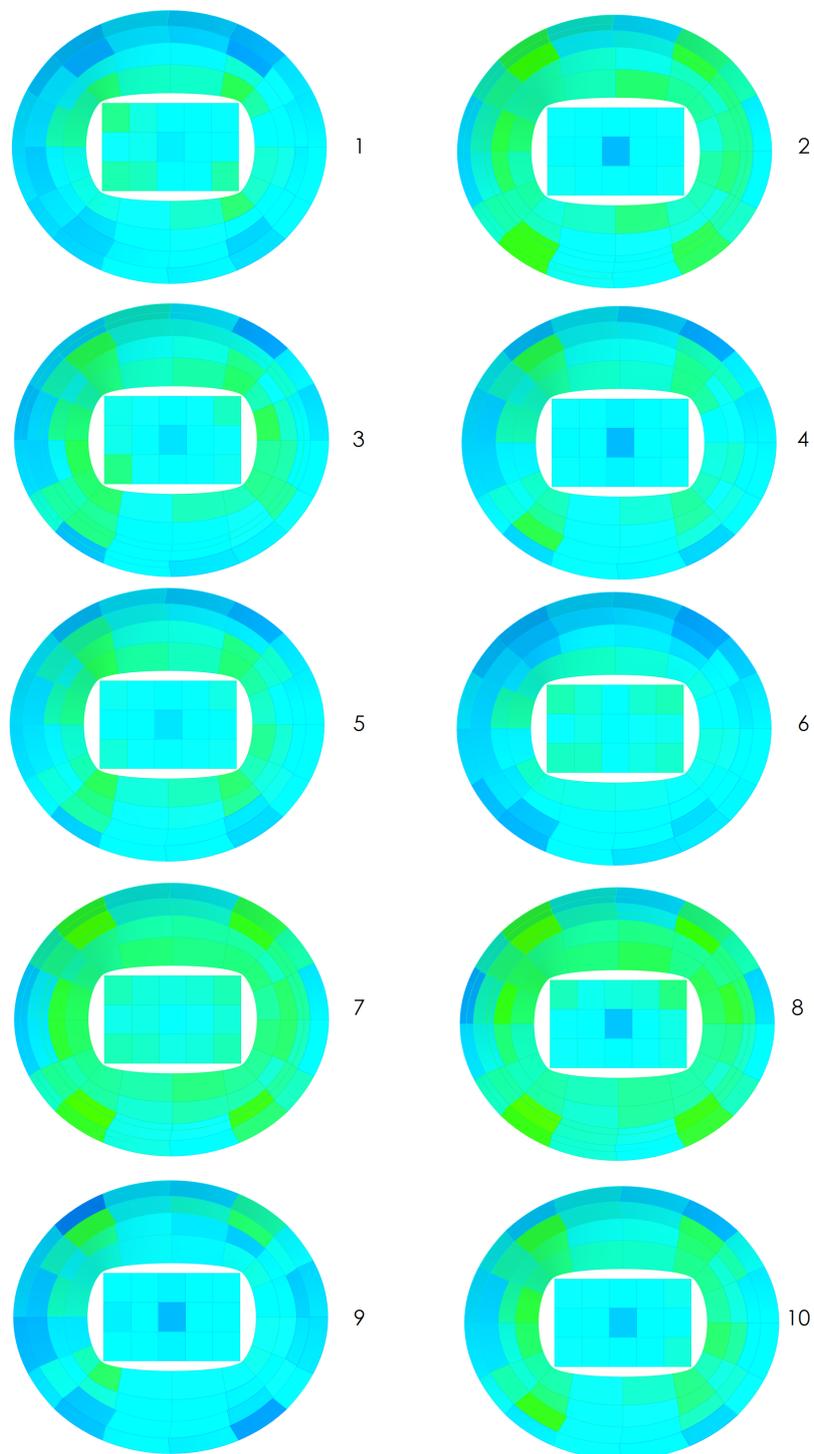


Figure 7.9: Acoustic result of each of the 10 validated results

8

Discussion

In this chapter, the results and outcome of the research are discussed. The findings are divided in two groups, namely the optimisation approach and the case study.

8.1 Optimisation approach

The starting point of this research was to develop a suitable approach for applying multidisciplinary design optimisation in the AEC industry and to investigate its potential. To decide whether or not an approach is suitable for applying on a relevant building design problem, several requirements were formulated (see Section 2.4).

According to these requirements, an approach within Rhinoceros and Grasshopper was chosen. The building design problem is to be set up parametrically in Grasshopper, after which the plugin Octopus drives the optimisation process. The evaluated solutions are passed to the custom post-processing tool PPOR, in which the optimisation results are to be investigated by the user to come to the desired solution.

This approach was then tested on the described case study; the structural and acoustic performance were evaluated for a large span steel stadium roof. After this case study, the performance of the proposed approach can be evaluated. This is done according the formulated requirements.

First of all, the required computation time is evaluated. For the case study, Octopus was able to evaluate 1.116 solutions in 32 hours. However, ideally such an optimisation process would be run overnight so that users do not have to wait for the results. In this case, the optimisation ran

8. Discussion

for 4 business days. On top of that, it showed that 1.116 evaluated solutions were not sufficient for estimating the Pareto front accurately, as some Pareto solutions were still improvable. This means that even more solutions need to be evaluated to accurately estimate the Pareto front, which could possibly cause a delay in the design process. However, in a traditional design process, the average duration of a design iteration is over one month (Flager and Haymaker, 2007). This shows that the applied approach does lead to a significant improvement, but that the computation time potentially could be reduced even further (see Chapter 10). Note that setting up the model also can take a notable amount of time.

Secondly, the approach was to be useable by architects and engineers themselves and not solely by researchers and developers. As it was chosen to keep the approach within Rhinoceros and Grasshopper, the user is required to have knowledge about these applications. When this knowledge is present, sufficiently to set up the building design problem parametrically, the different components (Octopus and PPOR) of the approach are not complex in use. As architects and engineers are (becoming more) familiar with Rhinoceros and Grasshopper, the approach should be usable for this target group. This could however be further investigated among architects and engineers.

Next requirement was the capability of the approach to post process the results, enabling the user to investigate the trade-offs, compare different results and find a desired solution. The custom post processing tool PPOR was developed, as Octopus appeared to have some deficiencies. PPOR enables the user to see the design associated with a solution, evaluate the results with different scatter and parallel plots and helps to filter the results. It helps the user to investigate the trade-offs between different objectives and influence of different variables, allowing the user to assess different solutions and better understand the design problem. However, when the number of objectives is bigger than three, scatter plots are not able to show the trade-offs between all objectives. In these cases, the trade-offs should be investigated with the help of a parallel plot.

Finally, the approach had to be applicable for different disciplines and design problems. As there are several plug-ins available for Grasshopper, enabling the analysis of different disciplines, the proposed approach is applicable for different disciplines and design problems (see Section 4.5). Furthermore, it is possible to create custom analysis tools (such as the SPL-Estimator) in Grasshopper rather easily. Analysis software 'outside' Grasshopper could also be used, as long as the performance indicators are fed back to Grasshopper. Otherwise, Octopus is not able to assess the solutions. As the ideal organisation of the different analysis components is heavily dependent on the problem, an investigation to find the ideal MDO architecture is recommended before the actual optimisation is conducted. This could have a large influence on the outcome of the optimisation process, as can be seen in Figure 4.4. It should furthermore be noted that in more complex cases as the researched case study, the computation time could increase even further. It is therefore recommended to further investigate the applicability of the approach (see Chapter 10).

It shows that the proposed approach complies with most of the formulated requirements. However, it has one main drawback, which is the computation time that is needed for evaluating different solutions. This is something that needs further investigation (see Chapter 10).

8.2 Case study

To investigate the applicability of the proposed approach, a case study was conducted. This case study focused on the acoustic and structural performance of the steel roof of the Donbass Arena. The acoustic deviation throughout the stadium was measured with the help of a custom ray tracing tool, called SPL-Estimator. The Karamba plug-in for Grasshopper was used to assess the structure and to measure its mass.

Although the optimisation process was not able to evaluate sufficient solutions to make an accurate estimation of the Pareto front, it was able to optimise the structural and acoustic performance by 58.3% and 22.8% respectively. Next, PPOR was able to find 72 designs which outperformed the baseline design for all objectives. 6 of these designs were Pareto optimal. It is however important to realise that these solutions are limited by the set-up of the used model, as for example the choice for variables, restraints and objectives is arbitrary. If the model is set up differently, other 'optimal' solutions are found. Also the made simplifications (see Section 5.2) have an influence on the found results.

Furthermore, PPOR helped investigating and understanding the influence of different design variables. It showed that especially the in the baseline design chosen slope of the roof and the number of divisions in the cantilevering trusses carrying the roof led to an inferior solution in terms of acoustic and structural performance. With these insights, it is possible to further optimise certain solutions by hand.

It should however be noted that the accuracy of the analysis methodologies could lead to some deviations in the results. The acoustic deviation is overestimated for solutions which acoustically behave worse than the baseline design. For solutions that acoustically outperform the baseline design, the acoustic deviation is underestimated. The structural mass is overestimated for all solutions, as Karamba showed to be conservative. For both the structural and acoustic performances, an error up around 5-10% should be assumed for the extreme solutions (see Section 5.2).

With these deviations in mind, it should be noted that this approach is mainly applicable in earlier design stages as the analysis methods in later stages should be more precise. Previous research showed that MDO could be most valuable in the exploring of the design space and thus in these conceptual design stages (Uijtenhaak and Coenders, 2012; Yang et al., 2015). However, setting up such an extensive model and approach in these (time limited) design stages might sometimes be inconvenient.

9

Conclusions

In this chapter, the conclusions that can be drawn from the conducted research are presented. The following conclusions can be drawn from applying the proposed optimisation approach on the Donbass Arena case study.

- The optimisation results show that the approach as proposed in this research is able to find solutions outperforming the baseline design for both the structural and acoustic objective.
- The computation time needed to accurately estimate the Pareto front of the optimisation problem as described in the case study proves to be undesirably long, as the evaluation of a single solution took about 100 seconds.
- As some found Pareto solutions proved to be suboptimal, insufficient solutions were evaluated to make an accurate estimation of the Pareto front.
- The post processing tool developed in this research, PPOR, proves to be able to provide the designer with the trade-offs between different objectives, the influence of different variables together with a representation of the different solutions.
- PPOR is able to obtain the correct data from the optimisation results, as is proved by validation of these results.
- The optimisation approach as proposed in this research is not restricted to the described case study and is applicable for different design problems and disciplines when the set-up is changed accordingly.
- Figure 4.4 shows that the set-up of the optimisation problem has a significant influence on the results found by the optimisation algorithm.

9. Conclusions

From the optimisation results, also some conclusions can be drawn about the design of the Donbass Arena.

- The size of the roof opening is the most influential variable for both the structural mass of the roof and the acoustic deviation in the stadium.
- The slope of the roof as it is in the baseline design is suboptimal for both the structural and acoustic objective.
- The number of divisions in the main cantilevering trusses which carry the remainder of the roof structure is too big in the baseline design, as decreasing the amount leads to a decrease in mass.

With these conclusions in mind, an answer is given on the main research question. This question was: *'In what manner and to what extent could a multidisciplinary design optimisation approach be applicable and beneficial for a large span steel stadium roof design, focused on its structural and acoustic quality?'*

The approach as proposed in this research was investigated as methodology for applying the multidisciplinary design optimisation on the aforementioned design problem. The approach was set up within Rhinoceros and Grasshopper, making use of the plug-in Octopus to drive the optimisation process. The structural performance was assessed by the plug-in Karamba and the acoustic performance was assessed by a custom ray tracing tool SPL-Estimator.

The approach proves to be beneficial, as it was able to find solutions outperforming the baseline design for both the acoustic and structural objective. The approach furthermore enables the user to investigate the trade-offs between different objectives and the influence of different design variables. These insights potentially lead towards better informed decision making. The applicability of the proposed approach is however limited with respect to the computation time.

10

Recommendations

In this chapter, recommendations are given for potential further research. Suggestions are given for further investigating the potential of the proposed multidisciplinary design optimisation approach and further exploring the case study. Finally, suggestions are given for improving or expanding the developed tools.

- Research could be conducted on how to decrease the computation time needed to evaluate each design. This should however not be at the expense of the further functionality and the ease of usability of the approach. A possibility is to investigate the application of cloud computing in this approach, an area in which already research is conducted (Jansen et al., 2014).
- Applying the proposed approach on a different design problem with different disciplines could give insight on the influence of different analysis methodologies on the computation time.
- Testing the approach on a problem with tight coupling between different disciplines, with for example an iterative loop, could further examine its applicability for more complex design problems.
- Applying the approach on a problem with more than 3 objective functions could investigate the effectiveness and 'clearness' of the post processing tool PPOR.
- The usability of the proposed approach could be investigated among architects and engineers.

10. Recommendations

- It could be interesting to include more disciplines in the optimisation process of the Donbass Arena case study.
- The acoustic and structural performance of the Donbass Arena could be described more elaborate by including more objective functions, variables or performance indicators. Examples are to minimise the number of different cross-sections, assess the acoustic performance in the case of a concert or evaluating different structural 'systems'.
- The accurateness SPL-Estimator could be further improved, by extending the acoustic analysis (see Section 3.1).
- The computation time of the SPL-Estimator could be further investigated and possibly improved. Note that this could be done along with the previous recommendation, as the accurateness and needed computation are strongly coupled.
- Expansion of PPOR towards for example a web-application could further improve its usability and enables different users to assess the results simultaneously.

Bibliography

- Academic (2014). *Donbass Arena*. Available from: <http://fr.academic.ru/dic.nsf/frwiki/532411>. (Accessed: 10 May 2016).
- Arup (2015). *Arup, we shape a better world*. Available from: <http://arup.com>. (Accessed: 28 September 2015).
- Bader, J. and Zitzler, E. (2008). Hype: An algorithm for fast hypervolume-based many-objective optimization. *TIK-Report No. 286*.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Bradner, E. and Davis, M. (2013). *Design Creativity: Using Pareto Analysis and Genetic Algorithms to Generate and Evaluate Design Alternatives*.
- Bureau, A. (2005). *NCCI: Elastic critical moment for lateral torsional buckling*.
- CATT (2015). *CATT Acoustic*. Available from: <http://www.catt.se/>. (Accessed: 26 January 2016).
- CESDb (2013). *LTBeam*. Available from: <http://www.cesdb.com/ltbeam.html>. (Accessed: 12 February 2016).
- Christensen, P. W. and Klarbring, A. (2009). *An Introduction to Structural Optimization*. Springer Science and Business Media.
- Coenders, J. L. (2009). Parametric and associative design as a strategy for conceptual design and delivery to bim. *Proceedings of the IASS 2009 Symposium*.
- Coenders, J. L. (2011). *NetworkedDesign: next generation infrastructure for computational design*. Delft: VSSD.
- Davidson, S. (2016). *Grasshopper: Algorithmic modeling for Rhino*. Available from: www.grasshopper3d.com (Accessed: 2 June 2016).
- Echenagucia, T. M. et al. (2013). Nurbs and mesh geometry in room acoustic ray-tracing simulation. *Proceedings of AIA-DAGA 2013 conference*.
- Echenagucia, T. M. et al. (2014). Multi-objective acoustic and structural design of shell structures for concert halls. *Proceedings of the IASS-SLTE 2014 Symposium "Shells, Membranes and Spatial Structures: Footprints"*.
- Elorza, D. O. (2005). *Room acoustics modeling using the raytracing method: implementation and evaluation*. University of Turku.

Bibliography

- European Committee for Standardization (2005). *Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings*.
- FC Shaktar Donetsk (2015). *Donbass Arena*. Available from: <http://www.donbass-arena.com>. (Accessed: 10 May 2016).
- Flager, F. and Haymaker, J. (2007). A comparison of multidisciplinary design, analysis and optimisation processes in the building construction and aerospace industries. *24th International Conference on Information Technology in Construction*.
- Flager, F. et al. (2009). Multidisciplinary process integration and design optimisation of a classroom building. *ITcon*, 14, p. 595-612.
- Galea, Y. (2005). *NCCI: Elastic critical moment of cantilevers*.
- Gerber, D. J. and Lin, S. H. E. (2014). Evolutionary energy performance feedback for design: Multidisciplinary design optimization and performance boundaries for design decision support. *Energy and Buildings*, 84, p. 426-441.
- Gerber, D. J. et al. (2012). Design optioneering: multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm. *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design*.
- Geyer, P. (2009). Component-oriented decomposition for multidisciplinary design optimization in building design. *Advanced Engineering Informatics*, 23(1), p. 12-31.
- Geyer, P. and Beucke, K. (2010). An integrative approach for using multidisciplinary design optimization in aec. *Proceedings of the International Conference on Computing in Civil and Building Engineering*.
- Geyer, P. and Rueckert, K. (2005). Conceptions for mdo in structural design of buildings. *6th World Congresses of Structural and Multidisciplinary Optimization*.
- Gracey and Associates (2015). *Sound and Vibration: Definitions, Terms, Units, Measurements*. Available from: <http://www.acoustic-glossary.co.uk/>. (Accessed: 26 January 2016).
- IMSCAD Ltd. (2013). *Architecture, Engineering and Construction*. Available from: www.imscadglobal.com/industries-AEC.php (Accessed: 2 June 2016).
- Jabi, W. (2013). *Parametric Design for architecture*. Laurence King Publishing.
- Jack, H. (2013). *Sound Measurement and Control*. Available from: www.engineeronadisk.com (Accessed: 3 May 2016).
- Jansen, D. et al. (2014). Interactive distributed optimisation for multidisciplinary design. *Proceedings of the IASS-SLTE 2014 Symposium "Shells, Membranes and Spatial Structures: Footprints"*.
- John, G. et al. (2013). *Stadia: The populous design and development guide*. 5th edn. Routledge.
- Konak, A. et al. (2006). Multi-objective optimization using genetic algorithms: a tutorial. *Reliability Engineering and System Safety*, 91, p. 992-1007.
- Marcuzzi, F. (2009). *A simple multi-objective optimization problem*.

- Martins, J. R. R. A. and Lambe, A. B. (2013). Multidisciplinary design optimization: A survey of architectures. *American Institute of Aeronautics and Astronautics (AIAA) Journal* .
- MathWorks (2016a). *Evaluate design trade-offs and find optimal designs*. Available from: http://nl.mathworks.com/discovery/design-optimization.html?s_tid=gn_loc_drop. (Accessed: 5 January 2016).
- MathWorks (2016b). *Genetic Algorithm*. Available from: <http://nl.mathworks.com/discovery/genetic-algorithm> (Accessed: 2 June 2016).
- Miller, N. (2012). *LunchBox*. Available from: <http://www.grasshopper3d.com/group/lunchbox>. (Accessed: 22 February 2016).
- Nederlands Normalisatie-Instituut (2011). *National Annex to NEN-EN 1990+A1+A1/C2: Euro-code: Basis of structural design*.
- Neill, M. et al. (2009). Decision support tools for parametric design. *Proceedings of the IASS 2009 Symposium* .
- Oasys Limited (2016). *GSA Analysis*. Available from: www.oasys-software.com (Accessed: 31 May 2016).
- Owens, B. (2014). *Stadium Acoustics Pump Up the Volume*. Available from: <https://www.insidescience.org/content/stadium-acoustics-pump-volume/1609>. (Accessed: 24 January 2016).
- Pelzer, S. et al. (2014). Integrating real-time room acoustics simulation into a cad modeling software to enhance the architectural design process. *Buildings*, 2, p. 113-138 .
- Preisinger, C. (2013). Linking structure and parametric geometry. *Architectural Design*, 83(2), p. 110-113 .
- Preisinger, C. (2015). *Karamba User Manual for Version 1.1.0*.
- R. McNeel and Associates (2016). *Rhinoceros*. Available from: www.rhino3d.com (Accessed: 2 June 2016).
- Reddy, J. N. (2006). *Introduction to the Finite Element Method*. 3rd edn. McGraw-Hill Education.
- Ren, Z. et al. (2011). Multi-disciplinary collaborative building design - a comparative study between multi-agent systems and multi-disciplinary optimisation approaches. *Automation in Construction* .
- Rutten, D. (2010). *Evolutionary Principles applied to Problem Solving*. Available from: <http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>. (Accessed: 6 January 2016).
- Schmit, L. A. (1960). Structural design by systematic synthesis. *2nd Conference on Electronic Computation*, p. 105-132 .
- Uijtenhaak, T. and Coenders, J. L. (2012). Multi-disciplinary design optimisation via dashboard portals. *IASS-APCS 2012: From spatial structures to space structures* .
- Vennard, M. (2013). How do you give stadiums atmosphere. *BBC Magazine* .

Bibliography

- Vierlinger, R. (2012). *Octopus*. Available from: <http://www.grasshopper3d.com/group/octopus>. (Accessed: 4 January 2016).
- Vorlander, M. (2010). Performance of computer simulations for architectural acoustics. *Proceedings of 20th International Congress on Acoustics, ICA2010*.
- WebFinance Inc. (2016). *Optimization*. Available from: <http://www.businessdictionary.com/definition/optimization> (Accessed: 2 June 2016).
- Wikipedia (2015a). *Performance-based Building Design*. Available from: https://en.wikipedia.org/wiki/Performance-based_building_design. (Accessed: 23 July 2015).
- Wikipedia (2015b). *Ray tracing (physics)*. Available from: [https://en.wikipedia.org/wiki/Ray_tracing_\(physics\)](https://en.wikipedia.org/wiki/Ray_tracing_(physics)). (Accessed: 26 January 2016).
- Wikipedia (2016a). *Algorithm*. Available from: <https://en.wikipedia.org/wiki/Algorithm#Formalization>. (Accessed: 6 January 2016).
- Wikipedia (2016b). *Building information modeling*. Available from: https://en.wikipedia.org/wiki/Building_information_modeling. (Accessed: 2 June 2016).
- Wikipedia (2016c). *Donbass Arena*. Available from: https://en.wikipedia.org/wiki/Donbass_Arena. (Accessed: 10 May 2016).
- Yang, D. et al. (2015). Multi-objective and multidisciplinary design optimization of large sports building envelopes: a case study. *Proceedings of the IASS Symposium 2015*.
- Zitzler, E. et al. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2), p. 173-195.
- Zitzler, E. et al. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-Report 103*.

Glossary

AEC industry The AEC industry consists of three separate disciplines: architecture, engineering and construction. They work together to design and realize building projects (IMSCAD Ltd., 2013).

Arup International engineering firm (Arup, 2015), where this Master's thesis project has been performed.

Associative design The concept with associations exposed to the user provides the user with the ability to create and modify associations on objects which describe a portion on logic which can be processed. These associations can be simple input-output mappings, but can also occur as mathematical expressions (Coenders, 2011).

Building Information Modelling (BIM) Process of generating and managing building data during its life cycle (Wikipedia, 2016b). There is some unclarity about what the term BIM exactly entails (Coenders, 2011).

Computer Aided Design (CAD) The use of computer systems to help create and modify a design. The term mainly refers to two-dimensional models and drawings (Coenders, 2011).

Computational design Designing with the help of the computing power of computers and scripting.

Finite Element Analysis A numerical technique to approximate the global behaviour, which uses finite elements to discretise a continuous material and assigns equations to each finite element. In structural engineering is mainly used to analyse structures (Reddy, 2006).

Genetic Algorithms (GA) Search which imitate the process of natural selection to generate solutions, often used for optimisation problems (MathWorks, 2016b).

Grasshopper Parametric and associative modelling system, which runs within Rhinoceros. It was developed by Robert McNeel and Associates (Davidson, 2016).

Karamba In Grasshopper embedded finite element program to predict the behaviour of structures under external loading (Preisinger, 2013).

Multidisciplinary Design Optimisation (MDO) Multidisciplinary design optimisation is a field of engineering that uses optimisation methods to solve design problems including a number of different disciplines (Martins and Lambe, 2013).

Octopus Grasshopper plug-in for applying evolutionary principles to parametric design and problem solving (Vierlinger, 2012).

Bibliography

Optimisation Optimisation is the process in which design variables are altered to achieve the best measurable performance, with regard to chosen objectives and under specified constraints (WebFinance Inc., 2016).

Parametric design Process based on algorithmic thinking that enables the expression of parameters and rules that, together, define the relationship between design intent and design response. These parameters can be defined and modified by the user (Jabi, 2013).

Performance based design Design approach where the design is required to meet certain measurable or predictable performance requirements (Wikipedia, 2015a).

Post Processing of Optimisation Results (PPOR) Custom post processing tool developed for this research, which helps to post process the optimisation results.

Ray tracing Method for calculating the path of waves or particles through a system which can contain regions with different propagation speed, absorption characteristics and reflecting surfaces (Wikipedia, 2015b).

Rhinoceros Advanced 3D modelling application software (R. McNeel and Associates, 2016). It was developed by Robert McNeel and Associates.

SPL-Estimator Custom ray tracing tool developed for this research, which is used to estimate the sound pressure levels in a room model.



Custom ray tracing tool (SPL-Estimator)

Rhino and its plug-in Grasshopper were chosen to model the building and to perform the multidisciplinary design optimisation in. Within Grasshopper a ray tracing method is used to analyse the acoustics and in more detail the sound pressure levels. The pressure levels are determined with the help of the formulas mentioned in Section 3.1.2. Several ray tracer tools are available to use in either Rhino or Grasshopper. Most of them however appeared either too comprehensive or not functioning correctly when a large number of rays (over 10.000) were used. With the size of the stadium in mind, such an amount of rays is definitely necessary to be able to estimate the sound paths. That is why it was chosen to make a custom Python script to model the rays within Grasshopper. The script of the tool (SPL-Estimator), together with the required input parameters and generated output, is shown in pseudo code below.

reflecting surfaces
absorption coefficients of reflecting surfaces
target surfaces (audience)
source point(s)
start vector(s)
maximum length of ray
maximum number of bounces

Table A.1: Required input ray tracing tool

A. Custom ray tracing tool (SPL-Estimator)

```
while number of bounces < maximum number of bounces

  if number of bounces == maximum number of bounces
    draw potential ray path from source point, with maximum length
    in direction of startvector

    for surface in list of reflecting surfaces
      determine intersection point of potential ray path with surface

      if intersection
        append potential rays from source point to intersection point
        append potential ray length
        append potential reflecting surface

    if potential reflecting surfaces exist
      determine shortest potential ray
      select length of shortest ray
      select corresponding reflecting surface
      determine intersection point of ray with reflecting surface

    else
      no intersection
      distance to reflecting surface is infinitely large

    for surface in target surfaces
      determine intersection point of potential ray path with surface

      if intersection
        append potential rays from source point to intersection point
        append potential ray length
        append potential target surface

    if potential target surfaces exist
      determine shortest potential ray
      select length of shortest ray
      select corresponding target surface
      determine intersection point of ray with target surface

    else
      no intersection
      distance to target surface is infinitely large

  if intersection with reflecting surface and distance to closest
  reflecting surface < distance to closest target surface
    append line from source point to intersection point
    with closest reflecting surface
```

A. Custom ray tracing tool (SPL-Estimator)

remaining length = **maximum length** - length of line
remaining sound power = $1 * (1 - \text{absorption coefficient of reflecting surface})$
remaining maximum number of bounces = **maximum number of bounces** - 1
determine surface normal at intersection point
rotate line 180 degrees around surface normal
determine end point of rotated line
next vector = vector from intersection point to end point
point = intersection point

elif intersection with target surface
append **line** from **source point** to intersection point with closest target surface
break

else
append **drawn potential ray path**
break

else
draw potential ray path from **point**, with **remaining length** in direction of **next vector**

for surface in list of **reflecting surfaces**
determine intersection point of potential ray path with surface

if intersection
append potential rays from **point** to intersection point
append potential ray length
append potential reflecting surface

if potential reflecting surfaces exist
determine shortest potential ray
select length of shortest ray
select corresponding reflecting surface
determine intersection point of ray with reflecting surface

else
no intersection
distance to reflecting surface is infinitely large

for surface in list of **target surfaces**
determine intersection point of potential ray path with surface

A. Custom ray tracing tool (SPL-Estimator)

```
    if intersection
        append potential rays from point to intersection point
        append potential ray length
        append potential reflecting surface

    if potential target surfaces exist
        determine shortest potential ray
        select length of shortest ray
        select corresponding target surface
        determine intersection point of ray with target surface

    else
        no intersection
        distance to target surface is infinitely large

    if intersection with reflecting surface and distance to closest
    reflecting surface < distance to closest target surface
        append line from point to intersection point
        with closest reflecting surface
        remaining length = remaining length - length of line
        remaining sound power = remaining sound power * (1 -
        absorption coefficient of reflecting surface)
        remaining maximum number of bounces = remaining maximum
        number of bounces - 1
        determine surface normal at intersection point
        rotate line 180 degrees around surface normal
        determine end point of rotated line
        next vector = vector from intersection point to end point
        point = intersection point

    elif intersection with target surface
        append line from point to intersection point
        with closest target surface
        break

    else
        append drawn potential ray path
        break
```

appended ray sections
remaining sound power

Table A.2: Generated output by ray tracing tool

A. Custom ray tracing tool (SPL-Estimator)

The output is then used to further assess the acoustic performance. After calculating the ray paths and the remaining sound power in each ray, the following steps are followed.

1. The remaining power of all rays which intersect with a receiving surface are summed and divided by the area of the receiver to determine the sound intensity at each receiver. Note that sound intensity is defined as the sound energy passing per second through a unit area held perpendicular to the direction of the propagating sound waves (Gracey and Associates, 2015). This is however simplified in this calculation to spare computation time.
2. The sound pressure level is finally calculated for each receiver surface.

With the sound pressure level known for each receiver surface, the acoustic 'performance' can be determined. How the performance is assessed can be found in Section 5.1.2. Figure A.1 shows the SPL-Estimator Grasshopper components.

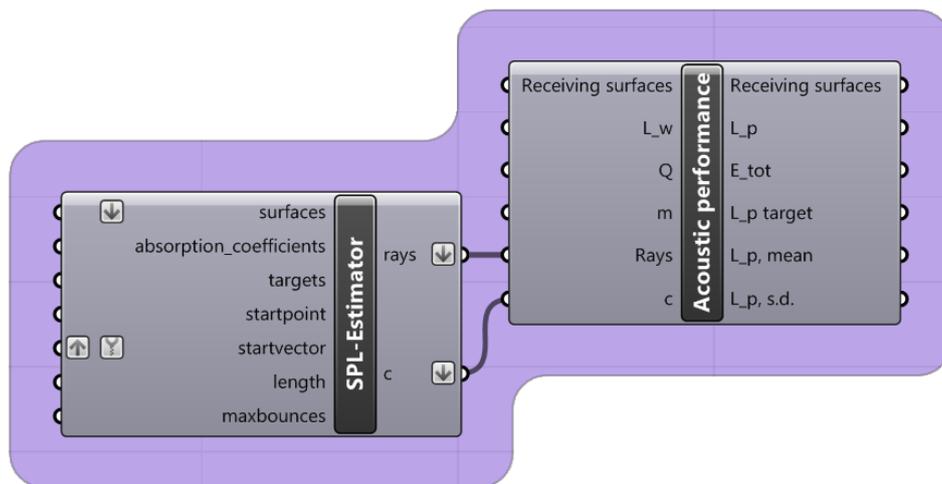


Figure A.1: SPL-Estimator Grasshopper components

B

Structural benchmarking calculations

Before applying the Karamba plug-in as structural analysis and optimisation for a large and complex structure, it is important to test the programme on simpler cases to investigate its potential inadequacies or limitations. The results found with Karamba will be compared to the results found with the structural analysis software GSA. The first four basic cases which are analysed are shown in Figure B.1. The comparison of the results are shown in Table 3.1 in Section 3.2.3. The results that are compared are the maximum bending moment and bending moment line (to verify the set-up of the models), maximum deflection, maximum stresses and maximum utilisation.

Although most results calculated with Karamba differ little from the results found with GSA, the maximum utilisation in first three cases are deviating significantly. Karamba is in all cases more conservative than GSA and an optimisation making use of this utilisation will thus lead to conservative and suboptimal designs. Further investigation shows that the deviation in utilisation is caused by differences in the elastic critical moment. This value is needed for calculating the lateral torsional buckling resistance according to Eurocode 3 (European Committee for Standardization, 2005) and for each of these cases, lateral torsional buckling is normative. However, the Eurocodes do not give a clear advice about how to determine this parameter. This explains the differences between the elastic critical moment and thus the maximal utilisation.

In this chapter, a more detailed comparison is made for each of these four cases.

B. Structural benchmarking calculations

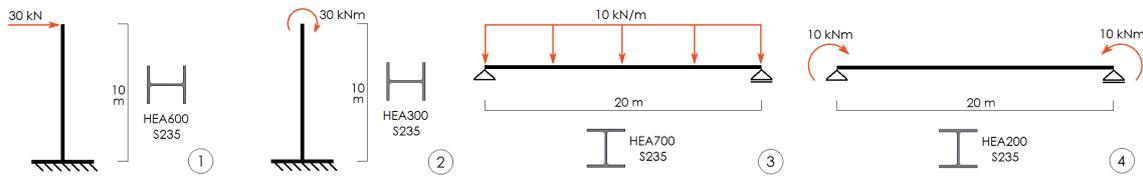


Figure B.1: Four basic structural benchmarking cases

B.1 Cantilevers - case 1 and 2

The first two cases are both cantilevers. The elastic critical moment is calculated as shown in Equation B.1 (Galea, 2005).

$$M_{cr} = C \cdot M_{cr,0}$$

$$\text{with : } M_{cr,0} = \frac{\pi}{L} \cdot \sqrt{EI_z GI_t}$$

$$\kappa_{wt} = \frac{1}{L} \cdot \sqrt{\frac{EI_w}{GI_t}}$$

$C = \text{loading coefficient}$

(SN006A Table 3.2 or Table 3.3)

(B.1)

and : $\eta = \text{destabilizing factor } (= 0)$

$E = \text{Young's modulus } (N/mm^2)$

$I_z = \text{second moment of area about the weak axis } (mm^4)$

$G = \text{shear modulus } (N/mm^2)$

$I_t = \text{torsion constant } (mm^4)$

$L = \text{beam length } (mm)$

$I_w = \text{warping constant } (mm^4)$

Next, the following factors should be calculated according to Eurocode 3 (European Committee for Standardization, 2005). With these factors known, finally $M_{b,Rd}$ can be determined.

$$\chi_{LT} = \frac{1}{\Phi_{LT} + \sqrt{\Phi_{LT}^2 - \beta \lambda_{LT}^2}}$$

(B.2)

$$\text{with : } \chi_{LT} \leq 1.0$$

$$\chi_{LT} \leq \frac{1}{\lambda_{LT}^2}$$

B. Structural benchmarking calculations

$$\Phi_{LT} = 0.5 \cdot (1 + \alpha_{LT}(\bar{\lambda}_{LT} - \bar{\lambda}_{LT,0}) + \beta \bar{\lambda}_{LT}^2)$$

with : α_{LT} = imperfection factor
 (Table 6.3 in Eurocode 3)
 $\bar{\lambda}_{LT,0} = 0.4$ (maximum value)
 $\beta = 0.75$ (minimum value)

and : $\bar{\lambda}_{LT} = \sqrt{\frac{W_y f_y}{M_{cr}}}$

$$\chi_{LT,mod} = \frac{\chi_{LT}}{f}$$

with : $\chi_{LT,mod} \leq 1$
 $\chi_{LT,mod} \leq \frac{1}{\bar{\lambda}_{LT}^2}$

and : $f = 1 - 0.5(1 - k_c)(1 - 2 \cdot (\bar{\lambda}_{LT} - 0.8)^2)$
 $f \leq 1.0$
 k_c = correction factor
 (Table 6.6 in Eurocode 3)

$$M_{b,Rd} = \frac{f_y}{\gamma_{M1}} W_y \chi_{LT,mod}$$

In Table B.1 and Table B.2, the calculations of both Karamba and GSA are shown. Values coloured in red (non conservative) or orange (conservative) are incorrect. In Section 3.2.3, it is explained why these values are wrong and what causes these errors.

B. Structural benchmarking calculations

	Karamba	GSA
E (N/mm^2)	210.000	205.000
G (N/mm^2)	80.769	78.850
I_z (mm^4)	$112.7 \cdot 10^6$	$112.7 \cdot 10^6$
I_w (mm^6)	$8.995 \cdot 10^{12}$	$8.995 \cdot 10^{12}$
I_t (mm^4)	$4.069 \cdot 10^6$	$4.069 \cdot 10^6$
l (mm)	$10 \cdot 10^3$	$10 \cdot 10^3$
$M_{cr,0}$ (kNm)	876	855
κ_{wt}	0.240	0.240
C	± 0.52	± 2.14
M_{cr} (kNm)	464	1829
W_y (mm^3)	$4.79 \cdot 10^6$	$5.35 \cdot 10^6$
f_y (N/mm^2)	235	225
λ_{LT}	1.56	0.81
α_{LT}	0.34	0.34
β	0.75	0.75
$\bar{\lambda}_{LT,0}$	0.4	0.4
Φ_{LT}	1.61	0.82
χ_{LT}	0.40	0.81
k_c	0.75	0.75
f	1	0.86
$\chi_{LT,mod}$	0.40	0.94
M_{Ed} (kNm)	300	300
M_{Rd} (kNm)	454	1130
unity check	0.66	0.27

Table B.1: Karamba and GSA check - case 1

B. Structural benchmarking calculations

	Karamba	GSA
E (N/mm^2)	210.000	205.000
G (N/mm^2)	80.769	78.850
I_z (mm^4)	$631.0 \cdot 10^5$	$631.0 \cdot 10^5$
I_w (mm^6)	$1.202 \cdot 10^{12}$	$1.202 \cdot 10^{12}$
I_t (mm^4)	$8.274 \cdot 10^5$	$8.274 \cdot 10^5$
l (mm)	$10 \cdot 10^3$	$10 \cdot 10^3$
$M_{cr,0}$ (kNm)	296	289
κ_{wt}	0.194	0.194
C	± 0.52	± 1.96
M_{cr} (kNm)	155	580
W_y (mm^3)	$1.26 \cdot 10^6$	$1.38 \cdot 10^6$
f_y (N/mm^2)	235	235
λ_{LT}	1.37	0.75
α_{LT}	0.34	0.34
β	0.75	0.75
$\bar{\lambda}_{LT,0}$	0.4	0.4
Φ_{LT}	1.37	0.77
χ_{LT}	0.49	0.84
k_c	1	1
f	1	1
$\chi_{LT,mod}$	0.49	0.84
M_{Ed} (kNm)	30	30
M_{Rd} (kNm)	144	275
unity check	0.21	0.11

Table B.2: Karamba and GSA check - case 2

B. Structural benchmarking calculations

B.2 Simply supported - case 3 and 4

The last two cases are both simply supported. The elastic critical moment is calculated as is shown in Equation B.6 (Bureau, 2005).

$$M_{cr} = C_1 \frac{\pi^2 EI_z}{(kL)^2} \left(\sqrt{\left(\frac{k}{k_w}\right)^2 \frac{I_w}{I_z} + \frac{(kL)^2 GI_t}{\pi^2 EI_z} + (C_2 z_g)^2} - C_2 z_g \right)$$

with : $E = \text{Young's modulus (N/mm}^2\text{)}$

$I_z = \text{second moment of area about the weak axis (mm}^4\text{)}$

$G = \text{shear modulus (N/mm}^2\text{)}$

$I_t = \text{torsion constant (mm}^4\text{)}$

$L = \text{beam length (mm)}$

$I_w = \text{warping constant (mm}^4\text{)}$

k and k_w are effective length factors

$z_g = \text{distance between loading point and shear center}$

C_1 and C_2 are loading and restraint coefficients

(B.6)

$$k = k_w = 1$$

$$C_2 = z_g = 0$$

(B.7)

$$M_{cr} = C_1 \frac{\pi^2 EI_z}{L^2} \sqrt{\frac{I_w}{I_z} + \frac{L^2 GI_t}{\pi^2 EI_z}}$$

The remainder of the calculation is identical to the first two cases. In Table B.3 and Table B.4, the calculations of both Karamba and GSA are shown. Values coloured in orange (conservative) are incorrect. In Section 3.2.3, it is explained why these values are wrong and what causes these errors.

B. Structural benchmarking calculations

	Karamba	GSA
E (N/mm^2)	210.000	205.000
G (N/mm^2)	80.769	78.850
I_z (mm^4)	$121.8 \cdot 10^6$	$121.8 \cdot 10^6$
I_w (mm^6)	$13.38 \cdot 10^{12}$	$13.38 \cdot 10^{12}$
I_t (mm^4)	$5.217 \cdot 10^6$	$5.217 \cdot 10^6$
l (mm)	$20 \cdot 10^3$	$20 \cdot 10^3$
$M_{cr,0}$ (kNm)	553	543
C	1	± 1.13
M_{cr} (kNm)	553	613
W_y (mm^3)	$6.24 \cdot 10^6$	$7.03 \cdot 10^6$
\bar{f}_y (N/mm^2)	235	225
λ_{LT}	1.63	1.61
α_{LT}	0.49	0.49
β	0.75	0.75
$\bar{\lambda}_{LT,0}$	0.4	0.4
Φ_{LT}	1.80	1.76
χ_{LT}	0.34	0.35
k_c	0.94	0.94
f	1	1
$\chi_{LT,mod}$	0.34	0.35
M_{Ed} (kNm)	500	500
M_{Rd} (kNm)	505	556
unity check	0.99	0.90

Table B.3: Karamba and GSA check - case 3

B. Structural benchmarking calculations

	Karamba	GSA
E (N/mm^2)	210.000	205.000
G (N/mm^2)	80.769	78.850
I_z (mm^4)	$133.6 \cdot 10^5$	$133.6 \cdot 10^5$
I_w (mm^6)	$10.82 \cdot 10^{10}$	$10.82 \cdot 10^{10}$
I_t (mm^4)	$2.105 \cdot 10^5$	$2.105 \cdot 10^5$
l (mm)	$20 \cdot 10^3$	$20 \cdot 10^3$
$M_{cr,0}$ (kNm)	35	34
C	1	1
M_{cr} (kNm)	35	34
W_y (mm^3)	$3.89 \cdot 10^5$	$4.30 \cdot 10^5$
f_y (N/mm^2)	235	235
$\bar{\lambda}_{LT}$	1.62	1.72
α_{LT}	0.34	0.34
β	0.75	0.75
$\bar{\lambda}_{LT,0}$	0.4	0.4
Φ_{LT}	1.69	1.84
χ_{LT}	0.38	0.34
k_c	1	1
f	1	1
$\chi_{LT,mod}$	0.38	0.34
M_{Ed} (kNm)	10	10
M_{Rd} (kNm)	35	34
unity check	0.29	0.29

Table B.4: Karamba and GSA check - case 4



Post processing tool (PPOR)

To show the actual output of the post processing tool, PPOR, an example is shown in Figure C.2. As stated earlier, PPOR is a custom tool used to post process the optimisation results. This tool is made in Grasshopper, thus the output is shown in Rhinoceros. The top left viewport presents the trade-offs between the two disciplines (structure and acoustics). At the top right, the trade-off shown between the roof opening (0) and the acoustic deviation (14). In the bottom left, the parallel plot of the solutions are shown and in the bottom right a the selected design solutions is presented. In all graphs Pareto optimal solutions are shown in orange, other feasible solutions in blue, infeasible designs are gray and the selected solution (in Grasshopper) is shown in green. The indices representing the different variables and performance indicators are clarified in Table 5.2 in Chapter 5.

Figure C.1 shows the Grasshopper component. Inputs are the designs, a Boolean toggle to choose whether all solutions need to be shown or only the Pareto solutions, the values of the variables, the selected parameters and the applied domain for these parameters, the number of value lists that are entered together with the number of objectives and finally the height of the output plots, the step sizes in the plot and the index of the design which need to be selected. Given output are the trade-off points together with the axes and captions, the selected design, the parameters describing the design and the parallel plot with its captions and indicated ranges.

C. Post processing tool (PPOR)

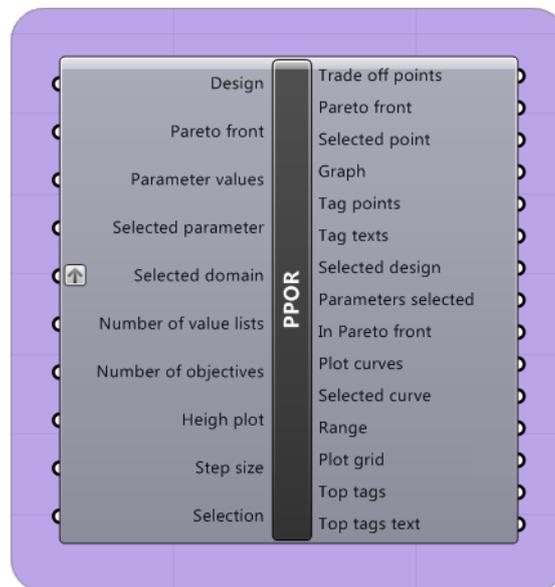


Figure C.1: PPOR Grasshopper component

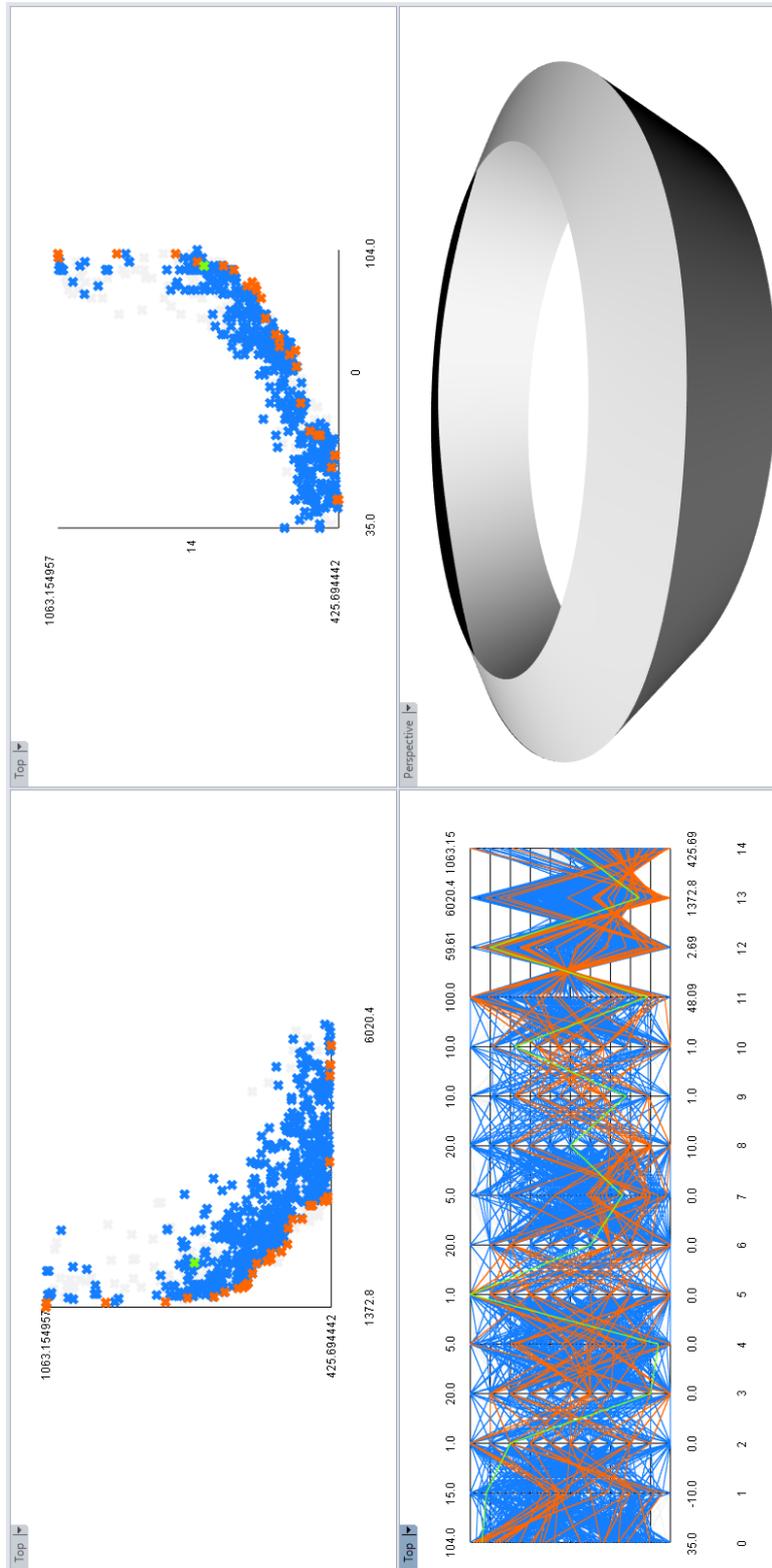


Figure C.2: Output of PPOR as shown in Rhinoceros



List of Figures

1.1	Flowchart illustrating the foreseen research methodology	18
2.1	Example of a Pareto front	23
2.2	Schematisation of described multidisciplinary system	25
2.3	Fitness landscape in evolutionary optimisation process (Rutten, 2010)	28
3.1	Ray tracing concept with 'S' as sound source	34
3.2	Reflection (specular and scattered) and absorption of rays	35
3.3	Test results with room shapes (first row), SPL-Estimator results (second row) and CATT results (third row)	38
3.4	Histogram of sound pressure levels test case 1 with average and standard deviation	39
3.5	Histogram of sound pressure levels test case 2 with average and standard deviation	39
3.6	Histogram of sound pressure levels test case 3 with average and standard deviation	40
3.7	Influence of the number of rays on SPL values (number of receiving surfaces is constant at 64)	41
3.8	Influence of the number of receiving surfaces on the SPL values (number of rays is constant at 20.164)	41

D. List of Figures

3.9	Structural optimisation classes	43
3.10	Example of a structural model in Karamba, with the loading and utilisation (red shows compression, blue shows tension) of the beam	44
3.11	Simple structural benchmarking checks	46
3.12	Karamba benchmark truss design	49
4.1	Example of Octopus set-up	52
4.2	Example of a Pareto front output from Octopus	52
4.3	Design for MDO test case	53
4.4	Results MDO set-up tests	54
4.5	Schematisation MDO architecture	55
4.6	Applied Design Structure Matrix	56
4.7	Example of a parallel coordinate chart	58
4.8	Output produced by the custom post processing tool PPOR	59
4.9	Comparison of the results in Octopus (left),the custom post processing tool (middle) and literature including the solutions from PPOR (right)	61
4.10	Schematic overview of the proposed MDO approach	62
5.1	Picture of Donbass Arena (FC Shaktar Donetsk, 2015)	65
5.2	Roof structure of Donbass Arena during construction (Academic, 2014)	67
5.3	Variables describing the Donbass Arena	70
5.4	Sensitivity analysis of the different variables	71
5.5	Outer shape of the model, with original Donbass Arena geometry (left) and altered geometry (right)	74
5.6	Structural model of the Donbass roof, with Karamba elements in blue	75
5.7	Indices of the checked normative cantilever	76
5.8	Histogram of the utilisation difference between Karamba and GSA for the normative cantilevering truss	78
5.9	Overview of the elements in the acoustic model	79
5.10	Model of the baseline design	83

6.1	Scatter plot of the feasible case study results (Pareto front in orange, unfeasible solutions in gray and baseline design in green)	86
6.2	Trade off between the acoustic and structural objective in Octopus (left) and PPOR (right)	86
6.3	Parallel plot of feasible solutions (Pareto solutions in orange, baseline in green)	88
6.4	Parallel plot of Pareto optimal solutions (Baseline design in green)	88
6.5	Scatter plot with solutions outperforming the baseline design (baseline indicated in green, Pareto optimal solutions in orange)	89
6.6	Parallel plot solutions outperforming the baseline design (baseline indicated in green, Pareto optimal solutions in orange)	90
6.7	Designs of the Pareto optimal solutions outperforming the baseline design for all aspects	91
7.1	Scatter plot with the 10 validated results (in green)	93
7.2	Scatter plot showing trade-offs between the roof opening, structural mass (left) and acoustic deviation (right)	94
7.3	Scatter plot showing trade-offs between the roof height difference, structural mass (left) and acoustic deviation (right)	95
7.4	Scatter plot showing trade-offs between the number of divisions in the trusses, structural mass (left) and acoustic deviation (right)	96
7.5	Scatter plot showing trade-offs between the structural height at the beginning of the cantilevering trusses, structural mass (left) and acoustic deviation (right)	96
7.6	Topview of the evaluated solutions, with the 8 source points (white), cantilevering trusses (black) and the sine shapes (blue and red)	98
7.7	Geometry of each of the 10 validated results	100
7.8	A cantilevering truss of each of the 10 validated results	101
7.9	Acoustic result of each of the 10 validated results	102
A.1	SPL-Estimator Grasshopper components	121
B.1	Four basic structural benchmarking cases	124
C.1	PPOR Grasshopper component	132
C.2	Output of PPOR as shown in Rhinoceros	133

E

List of Tables

3.1	Comparison outcome of simple benchmark cases	46
3.2	Comparison elastic critical moment of different simple benchmark cases (difference in comparison with LTBeam result)	47
3.3	Comparison between structural analysis results of Karamba and GSA for benchmark truss structure	50
4.1	Validation of 10 random solutions from the post processing tool PPOR	61
5.1	Donbass Arena facts	66
5.2	Variables of the Donbass Arena case study	69
5.3	Load combinations applied on roof structure	75
5.4	Comparison of the normal forces in the normative cantilever as determined by Karamba and GSA	76
5.5	Comparison of the support reactions of the normative cantilever as determined by Karamba and GSA	77
5.6	Maximum displacement of the normative cantilevering truss as found in Karamba and GSA	79
5.7	Absorption coefficients applied in the model	80

E. List of Tables

5.8	Acoustic values for three described designs as determined by SPL-Estimator and CATT Acoustic	81
5.9	Performances of the baseline design	83
6.1	Performance values of Pareto front solutions	87
7.1	Validation of 10 random solutions, manually reinstated in Grasshopper	99
A.1	Required input ray tracing tool	117
A.2	Generated output by ray tracing tool	120
B.1	Karamba and GSA check - case 1	126
B.2	Karamba and GSA check - case 2	127
B.3	Karamba and GSA check - case 3	129
B.4	Karamba and GSA check - case 4	130

