Communicating ‘structural’ design options

Using dashboard portals for exploring alternatives
Communicating ‘structural’ design options

Using dashboard portals for exploring alternatives

Tijl Uijtenhaak, c1049577
June 2011

Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Design & Construction
Section Structural Mechanics

BEMNext laboratory
Arup Amsterdam
This thesis consists of two parts; first there is an investigative study that focuses on the multi-disciplinary approach of building projects, second there is the design and development of a tool that has been developed for communicating alternatives in multi-disciplinary optimisation projects. Furthermore the tool brings out the information that would be valuable for a structural engineer to have presented in searching for the ‘optimal’ multidisciplinary solution.
Part A

Design is an ill-structured process that is hard to grasp, but is possible to structure in a way that the same steps lead to a result of the posed problem. The importance of mutual understanding between the different people who work on the same design problem is vital for success. By structuring the road to a solution and understanding the differences in (tacit) knowledge between the parties, a similar approach can be taken for all problems. Looking from a multi-disciplinary point of view at building design, optimisation is a very important part. When all disciplines work without interacting the total design would be a mixture of optimal solutions; leading to a sub-optimal ‘total’ solution because there will be no coherency.

To create coherency, parametric design provides the possibility to freely change all parameters within the previously set boundary. It creates the possibility to automate the process of changing all constraints and let the computer crunch through all options. This will generate the data needed for further examination. To go through all these options is too tedious; dashboard technology can solve this problem in a very graphical/visual way. Their strength is to output all the data and have multiple users understand in a glance where the focus areas should be. This will lead to better substantiated decisions where the relation between choice and consequence is much clearer. Besides that it provides the possibility to dive deeper into the data and extensively explore visual aspects (renders/videos), timelines or graphs (trends, pivoting points and black swans), which is very helpful in showing other parties where the highest profit can be expected.

To have useful data on the dashboard it is important to store data and then mine it, therefore databases are very important. If the databases are accessible via the internet the advantages can be even higher since when multiple disciplines are spread over different offices they can still access them. Professional databases make sure the data is stored safely and there is next to no downtime.

Part B

For the development of a tool it was important to apply the theoretical knowledge to a concrete example. Keeping in mind that design (in this case architectural) is a matter of taste. Besides showing that you understand the brief of the project developer, everything comes down to convincing that your plan will suit his wishes the best. To do this the design should not only satisfy the wishes but also fill the client with pride. An important issue here is that the subjective nature of taking decisions is fed by the information someone receives, “what we do is based on what
we know”. This is another trouble in multi-disciplinary optimisation, not all disciplines have the same skill nor same high quality tools for presenting their added value. This tool should help with decision making in the process of multi-disciplinary design optimisation by providing the same tool to every party. Convincing the other disciplines happens via increasing mutual understanding, or ‘sharing’ tacit knowledge, mostly through a visual environment. Once there is mutual understanding it is possible to change the (discipline individual) criteria and work towards an ‘optimal’ whole solution. In this tool designing is separated into three different phases:

1. Generating the alternatives
2. Searching through the alternatives
Figure S.2 Scheme showing the steps in the process of the usage of the tool.
In the first part of the design process a sketch will be made, after the concept is agreed upon a parametric model will be made. The Rhino ‘Architectural’ parametric geometry should be converted to data that can be implemented by analysis software.

Once all data is ordered and stored in a database (back-end) it should be structured in such a way that a program (front-end) can browse through the options. Via Django queries can be run on a HTML page and directly be searched for in the database.

These queries are organised from the dashboard and should in the future control the input for the database and provide a way to individually customise the dashboard via widgets and operators. The widgets should determine the layout of the dashboard and the operators determine the output of the dashboard. Every time a value in the operators is modified the content of the dashboard changes and shows the user a new dataset. By the changing of the operators the data can easily be explored and once the results are satisfying these can be exported to an automated report or stored as a preferred set. The next step could be to dig deeper into the data with the new insights and maybe with the use of different widgets.

Although the previous ideas are only outlook, a case study has been performed to see what kind of data would be interesting and what a diagrid tower could teach us as valuable information. A diagrid proved to be very useful, since all multidisciplinary aspects can be found and the relations between the topics are almost endless. Within this case study the focus was mainly on the structure. A hand calculation and a GSA calculation has been made to get a feel of the structure and find which parameters are important to bear in mind when working on the mock up dashboard. The main parameters that are governing are the angle and length of the diagonal and the area of the members in the diagonal.

With the tool development the focus was on building all the separate parts to have a test run. This should show if the dashboard could help making decisions based on, in this case graphs, the visual information displayed. The whole framework is coded in python and this controls the tool from generating the data via grasshopper, storing this in a database, to the Django framework outputting the template of a dashboard.

The generated data is tested via the Tool and the results are validated. As a result it can be stated that the program works very smooth, but it is a real engineer’s tool; you have to know what you are doing to browse through the data, errors can hide and wrong conclusions can easily be drawn. In this case the Salamander element (connecting Grasshopper and GSA and collecting the results) appeared
to have a few errors in it, therefore the results are not valid (structurally). The tool proved to be powerful in tracing the wrong data even with a large dataset. These results are very promising for further investigation and a good step towards a further elaborated multi-disciplinary optimisation decision tool.

Figure S.3 Screenshot of the developed dashboard.
This thesis is the result of an interest that I have developed through my studies of both Architecture and Civil Engineering. At first when I attempted to combine both master theses in a single, holistic project it seemed that the two were incompatible. The conflict of interests between myself as both the structural engineer and the architect was the main difficulty. On the one hand the design had to be of outstanding quality in space, tectonics, organization and an overall experience of integrated structure and design. On the other hand I wanted the structure to be rationalised and sound. And so from the outset I was faced with the problem of fusing two disciplines, each of which had separate priorities.

In the real world the involved parties are far more diverse, with there being more disciplines involved than just the architect and structural engineer. Ultimately it is often the case that a project budget will have the final say in the decisions made, but it should still nevertheless be remembered that the overall aim should be to achieve the highest possible quality. Therefore communication between the various disciplines involved is of uttermost importance, as well as being able to explain others parties why certain choices are made and how the design would benefit from this. This conclusion sparked my curiosity into multi-disciplinary projects and how solutions are found to a specific design problem. The objective in my work was to then take this to the next level, seeing if it is possible to integrate different disciplines and display the relation between choices and consequences in a graphical manner.

This project started off with many gaps in my knowledge about multi-disciplinary optimisation, coupled with a complete lack in programming experience. This led to my expectations at the outset of the thesis work being somewhat vague. The first part of the thesis work was mainly spent getting acquainted with Python, computer programming and algorithms in general. Having completed the thesis work whilst I am more familiar with programming language, I recognise that there is still a wealth of skills to learn. However, I sincerely believe that the future of the structural engineering will be ever more intimately linked with developing and coding your own programs.

Overall thesis subject has developed into a professional interest, which I now hope to be able to further explore based on my recommendations for further study.

This thesis project would not have been possible without the support of my graduation committee. Furthermore I would like to thank Arup Amsterdam for the opportunity the office gave me to combine my thesis with working on live projects, allowing me to gain further insight into the concept of “total architecture”.
<table>
<thead>
<tr>
<th>PART</th>
<th>A</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Design Process</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>1.1 Reflection in Action Model</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>1.2 Rational Model</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>1.3 Sharing knowledge</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>1.4 Creating knowledge</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>1.5 Refined design and knowledge creation in the building sector</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>1.6 Conceptual/preliminary design</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>1.7 Final draft</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>1.8 Post-construction design</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>1.9 Conclusion</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Multi-Disciplinary Optimisation</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>2.1 Optimisation</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>2.2 Interactive optimisation</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>2.3 Design problems</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>2.4 Setting up a Multi-Disciplinary Optimisation project</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>2.5 Conclusion</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Parametric Design</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>3.1 Parametric and associative design</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>3.2 Constraint based</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>3.3 Associative combination</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>3.4 Relation based</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>3.5 Conclusion</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>The Dashboard</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>4.1 Dashboard Layout</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>4.2 Conclusion</td>
<td>41</td>
<td></td>
</tr>
</tbody>
</table>
11.6 Dashboard  131
11.7 Testing and results  132
11.8 Validating the results  135
11.9 Evaluating the results  149

Conclusions, Recommendations And Evaluations  153

12.1 Conclusion  155
12.2 Recommendations  156
12.3 Evaluation  158

Bibliography  161

Appendix  165

Cantilever-beam, Bending And Shear  A1
Braced Frame  B1
Part Of The Diagrid  C1
validating The Tower  D1
Graduation Committee & Personalia Author  E1
This thesis consists of two parts; first there is an investigative study that focuses on the multi-disciplinary approach of building projects, second there is the design and development of a tool that has been developed for communicating alternatives in multi-disciplinary optimisation projects. Furthermore the tool brings out the information that would be valuable for a structural engineer to have presented in searching for the ‘optimal’ multidisciplinary solution.
The building design process is a complex one with many disciplines involved, whilst at the same time being driven by the wishes and ambitions of the client. Traditionally a design team would consist of architects and structural engineers. Nowadays however design teams will often include other areas of expertise, such as urban planners, landscape-architects, along with environmental and building services engineers.

Such design groups are responsible for the overall designs, but then the following building processes are conducted by construction professionals, who are in turn supported by different specialist sub-contractors. It is important for this reason that the construction team members are also able to understand the reasoning behind the decisions taken by the design group, in order to then be able to make the designs into a reality.

However it is also expected that at least one of the parties involved in the process to be the overall coordinator of the design and construction. This party, or parties, needs to have a clear overview of the process itself, whilst at the same time have sufficient overall knowledge to be able to converse effectively with the other disciplines.

“Ever since Vitruvius’ first century treatise on architecture, we accept axiomatically that a designer must know a little bit about everything because design work requires varied knowledge and an outstanding capability form mental integration and synthesis. (Gabriela Goldschmidt, 1995)

Historically speaking the various parties involved tend to be independent specialists. They then attempt to integrate their respective skills to reach a holistic and satisfying solution. This often leads to misunderstandings and inefficiencies in achieving the required result, leading to work being completed neither on time nor within budget. Not only is miscommunication a problem, but sometimes there also exists a lack of trust between the different parties involved. Misunderstandings for example will often lead to the urban planner
become distrustful of the architect, whom may in turn have their reserves about the project developer, who doesn’t trust the contractor, etc. The building sector starts to become a chain of distrusting parties creating a phenomenally disorganised clutter.

“**In our work as structural engineers we had - and have – to satisfy the criteria for a sound, lasting and economical structure. We add to that the claim that it should be pleasing aesthetically, [...]. And then we come up against the fact that a structure is generally a part of a larger unit, and we are frustrated because to strive for quality in only a part is almost useless if the whole is undistinguished, unless the structure is large enough to make an impact on its own. We are led to seek overall quality, fitness for purpose, as well as satisfying or significant forms and economy of construction. To this must be added harmony with the surroundings and the overall plan.** (Ove Arup, 1970)

It is in the multi-disciplinary design space that there is a lot of improvement possible. It is necessary to understand the relations between constraints and their consequences, thereby adding value through the cooperation achieved with the designing party and structural engineer. This process enables well founded decisions to be made, which take into account the information supplied by the various disciplines involved.

The question that arises is; “Is it possible to design a computational tool that supports multi-disciplinary optimisation, doing this by providing a platform upon which any of the experts involved are able to see their influence on the total design process?” A second question that comes up is; “How can one judge the importance or added value of each discipline?” These two questions lead to the main research objective:

The aim is to design a program with which parametric design can be evaluated (by an expert), and even on a multidisciplinary level can be used to reach the ‘most desirable’ solution, while having an understandable range of solutions in which the criteria and relations guide one’s gut feeling in selecting the outcome. This range will then bring common tendencies of understanding of the criteria to light, and the effects the relations have concerning the outcome.

This research objective is then extended by two sub objectives that can be read in part B of chapter 2.

To find the answers first a literature study (Part A) is executed to become more familiar with multi-disciplinary projects, how parties cooperate, which tools are
available to store and present data and to acquire skills in programming. Following this in Part B the research objective is extended and built upon with the knowledge gained in Part A. A design for a tool is then developed that could help in multi-disciplinary optimisation. After validating the first results via a case study, the information gained serves in carrying out improvements to the tool more accurate and in re-adjusting the focus of the thesis. In the end of part B the development of the tool is explained. After this an evaluation is written of the process from which the recommendations for further study are then suggested.

This thesis can be seen as the first step in a new direction of multi-disciplinary optimisation. A concept is laid out that could be implemented and this is explored, with the main focus on the role of the structural engineer, where data is generated automatically and is shown in a visual manner on a dashboard. Furthermore, there are many extensions possible that could lead to highly advanced multi-disciplinary optimisations. For example, elaborated algorithms could speed up the process and extend the interactivity capabilities.
PART A

LITERATURE STUDIES
Before building the tool, extensive research should be done to fully comprehend what is needed and to prevent reinventing the wheel. In this chapter the focus is on communication between different disciplines, how do different disciplines work together on large complicated building projects and what information do they need from each other to get to a better coherent design, leading to a higher valued outcome. To get to this, knowledge should be shared but in what kind of way and how much should the other discipline know, how can this be presented in a way that all parties understand the information given by the other parties as well as all information is presented in such a way that it is equally important.
Designing is a complex act, seldom fully understood (even when in the process itself), this causes a kind of mystique that designers/artists like to create. Especially the conceptual design part, which is the most crucial part in the engineering product development cycle, is very intriguing. It can be seen as an iterative process in which multiple parties can be involved. All parties have different skills and different tastes therefore they will influence the design in a specific way. Because it is hard to understand how a design process works, there exists substantial disagreement which resolves in several theories describing the process. In general there are two basic but fundamentally different ways: one is the **Rational Model** and the other is the **Reflection in Action Model**\(^1\).

In this thesis the approach of the Rational Model will be followed. Although a brief exploration of the Reflection in Action Model will be given to understand the difference in approach. The Reflection in Action Model is a label given to a collection of interrelated concepts.

### 1.1 Reflection in Action Model

The main idea behind the Reflection in Action Model (developed by Schön 1983) is that:

Every design problem is unique, since the ambition, location and budget of every design problem is different. Thereby the design team will differ in structure and the focus within the group will shift depending on the persuasiveness of the team members. One of the talents of the experienced designers should be to know how to tackle a design problem and work with the group, although never dealt with before. The design team will set its direction and come up with the approach of the problem at hand. Every team will have a different approach and depending on the character of the persons a design specific solution will be improvised on to take action and improve the current situation \(^2\).

This can be translated to the following:

- designers use creativity and emotion to generate design concepts,
- There is no set sequence of design steps via improvised manners steps as analysis, design and implementation are contemporary and inseparably linked.

### 1.2 Rational Model

The Rational Model (developed by Simon in 1969), is that ratio will be used to come up with solutions for the design problem; therefore the logic will be similar to the positivistic framework of science which is used for the classical sciences such as physics. Set steps to conduct can be come up with and the same process

---

\(^1\) Dorst and Dijkhuis, 1995  
\(^2\) ibidem
can be used for different design problems. Much attention is paid to the rigour of the analysis, and the generic or objective nature of the process. Simon quotes optimization theory as a prime example of what he believes a science of design could and should be. And this can be translated in the following:

- Designers attempt to optimise a design candidate for known constraints and objectives;
- The design process is plan-driven;
- The design process is understood in terms of a discrete sequence of stages.

This very logical and structured approach of designing would probably not suit the situation as most designers, in different fields of expertise, see the design process. One of the reasons is that the goals are uncertain at the start since the requirements and constraints keep shifting during the design process. Sometimes knowledge gained through experience (tacit knowledge) make people believe they know how the process should be dealt with, but during the process uncertainties might pop up changing the whole process. Every time more knowledge is gained about the design, changes will be made and some will influence all the previous work. But if the process is highly abstracted this could be seen as the steps taken.

The reason of not following the approach of the reflection-action model is not that it is inaccurate or incorrect. But the aim of this thesis is to find a solution to automate a multi-disciplinary design process in such a way that the results can be presented in a clear and compact visual way to easily draw conclusions. If the approach of the reflection in action process was taken as a starting point, every design query would be different and there would be no use of trying to automate the process. On the other hand it could be seen that every step within the rational model could exists of a reflection-action set of steps.

<table>
<thead>
<tr>
<th></th>
<th>Rational Problem Solving</th>
<th>Reflection in Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designer</td>
<td>Information processor (in an objective reality)</td>
<td>Person constructing his/her reality</td>
</tr>
<tr>
<td>Design Problem</td>
<td>Ill defined, unstructured</td>
<td>Essentially unique</td>
</tr>
<tr>
<td>Design Process</td>
<td>A rational search process</td>
<td>A reflective conversation</td>
</tr>
<tr>
<td>Design Knowledge</td>
<td>Knowledge of design procedures and 'scientific' laws</td>
<td>Artistry of design: when to apply which procedure/piece of knowledge</td>
</tr>
<tr>
<td>Example Model</td>
<td>Optimisation theory, the natural sciences</td>
<td>Art/ social sciences</td>
</tr>
</tbody>
</table>

Table 1.1
The rational problem solving paradigm and the reflection-in-action paradigms summarised (Dorst and Dijkhuis, 1995).
1.3 Sharing knowledge

Most design queries are becoming more complex and therefore different disciplines work on the same design. So besides the design process itself multidisciplinary design has to do with sharing knowledge and getting to understand what knowledge the other involved parties contributed. This could be a problem since not often people are aware of the knowledge they possess or how it can be valuable to others. If multiple people are in a design process together it isn’t necessary to have the same knowledge, but it is very important to know whom and when to ask. To gain this knowledge transferring of (tacit) knowledge between different parties in the design process is of great importance.

Tacit or implicit knowledge (as opposed to formal or explicit knowledge) is knowledge that is difficult to transfer to another person by means of writing it down or verbalising it. It is knowledge expressed or understood without being directly stated. Tacit knowledge can been described as “know-how”, opposed to “know-what” (facts), “know-why” (science), or “know-who” (networking) [4].

In the building sector, everyone is trained in a different way and all the experience gained from previous projects is knowledge that is tacit. The process of transforming tacit knowledge into explicit or specifiable knowledge is known as codification, articulation, or specification. The tacit aspects of knowledge are those that cannot be codified, but can only be transmitted via training or gained through personal experience. Again the aim should not be to know everything the other knows, but to be acquainted with his expertise and know when to ask for assistance (“know-who”). Effective transfer of tacit knowledge generally requires extensive personal contact and trust, therefore at the start and during the process of a project, time should be spent to get to know each other and become familiar with everyone’s specialities.

1.4 Creating knowledge

A knowledge creation process as described by von Krogh et al (2000) is helpful in defining the steps to be taken by the team. According to them this process can be broken down to a number of stages:

1. Sharing tacit knowledge,
2. Creating a concept,
3. Justifying the concept,
4. Building a prototype,
5. Cross-levelling knowledge.

When these steps are taken the process is structured in a logical way and all parties can pull their weight in the process.
1.5 Refined design and knowledge creation in the building sector

Looking back at the steps described by von Krogh et al (2000) from the rational model point of view, these may have a few shortcomings. One of the shortcomings could be the ill-structured way of design assignments, causing trouble in following a strict procedure.

“Design problems are almost always new to some degree, and we therefore think of them as belonging to the category of ill structured problems, that require productive process to be adequately solved.” (Gabriela Goldschmidt 1995)

From the above it can be seen that design as a process is complicated, but it is possible to extend the above mentioned steps and make it a little more suitable for design processes in the building sector. To do this we first divide the design into three different stages, conceptual/preliminary design, final draft and post-construction design.

These stages could then be further elaborated. In this thesis the main focus will be on the conceptual design phase. The other phases are further down the line and for this thesis of less importance, since there is less flexibility in these phases and fewer decisions to be made. Therefore the computational advantage that could be gained via a tool is smaller.

The different design stages are further elaborated in the next sections.

1.6 Conceptual/preliminary design

To use the reflection in action model in building design assignments a set of steps could be defined that guide the process. These steps can be followed on every occasion but sometimes the focus on the different aspects will be altered. The steps are:

1. Begins with the design brief, an early (often the beginning) statement of design goals,
2. Analysis of current design goals from the design brief,
3. Investigating similar design solutions in the field or related topics,
4. Specifying requirements of a design solution for a product (program of requirements) or service,
5. Problem solving, conceptualising and documenting design solutions,
6. Presenting design solutions,
7. Cross levelling knowledge with all involved parties,
8. Preparation for the next Phase

These steps can be used to start with any design problem in a structured way.
The big advantage is then that a certain repetition will be introduced in tackling design problems, making it easier to work together. The same holds for new processes, although the outcome is not yet there, the roadmap is there to guide through the process. Therefore understanding the problem via a thorough analysis is an influential part of the design process.\footnote{Nigel Cross 1995}

This explains why the first four steps in the above mentioned are all about gaining insight in the design assignment. Sometimes the real challenge is in knowing what is asked for instead of coming up with an extraordinary design, which could be a solution to several issues. An interpretation of the phases above with specific reference to building projects is given in the following sections.

**Design brief**
At the beginning of a design process, the design brief is provided by the client. Sharing tacit knowledge starts at this point, by reading through the design brief carefully. The goal here is to have the focus of the brief understood by using both one’s own knowledge and that of one’s peers. In this phase it is also very important to understand the different roles and relationships within the team (seniority or experience, leader etc.).

**Analysis of design goals**
The creative process might be designing in itself, but this is more embodied in knowing what is being asked and where the focus should be. Reading the requirements and coming up with extra requirements to gain extra feeling for the project is an example of tacit knowledge in its purity occurring. Besides the analysis of the content of the design question it is very important to get a grasp of the amount of time needed to come up with solutions. A schedule should be made of how much time can and will be spent on the matter.

**Research**
After fully comprehending what is being asked, it is wise to do some historical research, finding precedents that had to deal with similar design questions. In a group gathering information is a more overt activity that has to be done to enhance mutual understanding when collaborating. Relevant information has to be gathered, but also extracted from its source and shared within the team. With the found precedents a concept can be formed that plays with the requirements and broadens the ideas behind it. The concept concerns more than just requirements of one of the disciplines; it has to work with multidisciplinary disciplines. The concept spoken of here is still more an idea of getting a feel of the design query and extending the knowledge of the question asked.
Specification
Coming from different points of view and backgrounds, each concept that is generated should be tested and lessons should be learned from the different ideas put forward. It is possible that some of these can be combined and a final list of requirements emerges that represents the best combination of the previous steps. This list of requirements should then be the reading of de design brief by the design team and all the shortcomings or extra points of attention needed according to them. Common understanding of the project should be the goal.

Conceptualising
Here “designing” starts. Since now there should be full apprehension of the design question, generating new ideas solving the issues and jointly build these concepts into a specific design proposal is the next step. This process is a whole process in itself and hard to catch in a program, and maybe more according to the reflection in action model. Much of the design activity (especially in the conceptual stage) is unplanned, intuitive and ad hoc. The drifting and continuous changing between options make it hard to understand what is done and how the ‘solutions’ came to the table. The team will develop initial concepts into more detailed and robust versions. After that the team has to decide if aspects of the other ideas can be adopted and added to the solution at hand to come to an even better solution.

Presentation
The phase after justifying a concept and working up to a robust prototype is an important one, since here it is proven whether all the conceptual ideas are still valid and if it is possible for the concept to be realised. The prototype will reveal all the disciplines involved and should lead to a deeper understanding of the “do’s” and “don’ts”. After the prototype has been built and the initial testing phase has been carried out; others can give their opinions and customers or other parties that will eventually be involved can share their views. This is because now all the tacit knowledge has been converted to a product that can be assessed.

Cross levelling
After the presentation there will be cross levelling with all the parties involved. This should be done to clarify that the reading of the brief is according to the wishes of the client. This does not mean that there will be no interaction during the other phases, but only after making the concepts concrete and having an extended presentation with drawings, models, schemes and an explanation it can be assumed that everyone knows what is going on. After the presentation there will be questions and maybe even some wrong chosen assumptions. This would be the time to calibrate the design
process and go confidentially into the next phase.

**Next Phase**

After the presentation and cross levelling the next phase can be started. All the concepts can go to a more detailed level and knowledge gained in the previous phases will be exploited. The design will be extended with new ideas and previously ignored solutions. This will go on till the next presentation, cross levelling phase and eventually to the next status.

The next sections, although very important in the building process, are only lightly touched and not explained into depth, since this thesis is focussing on the (conceptual) design phase.

### 1.7 Final draft

The final draft is the phase in which the last test will be executed concerning buildability and how these effect the design in detailing or if some ideas are too expensive and should be changed. Besides that the final plans and sections should be designed and all the disciplines should be integrated even more into the whole.

- Development -continuation and improvement of a designed solution;
- Testing -in situ testing a designed solution.

During construction it is of uttermost importance to stay flexible and think with all the gained knowledge in the process of building of what is coming. If errors in a mock up appear, they should be taken care of and made sure that the trouble is solved before continuing.

### 1.8 Post-construction design

After a building is build a lot of information can still be gained by evaluating the building via feedback from all parties in the design as well as the contractor, client and most importantly the people using the building. Besides that if the building is built in a new to develop area, most likely the rest of the area still needs finishing, either via building or growing. The users using the area and buildings have to get familiar with the new situation.

- Implementation -introducing the designed solution into the environment;
- Evaluation -summary of process and results, including constructive criticism and suggestions for future improvements
In the phase after the design has been realised a very important phase (the longest phase off all) starts. Here all the design concepts can be verified and validated. If the client and the designers make the effort to measure what is happening they can gain an enormous amount of data that can be of high value in future designs.

1.9 Conclusion

The previous section showed that designing is an ill-structured problem that is hard to grasp, but is possible to structure in a way that the same steps lead to a result of the posed problem. The importance mutual understanding between the different people who work on the same design problem is vital for success. By structuring the road to a solution and understanding the (tacit) knowledge differences between the parties a similar approach can be taken for all problems. This study will help keeping the focus of the tool on the importance of the broad variation between possibilities leading to success, and the importance of an agile tool that can vary when the design query needs it to vary, but still have a set approach that guides the process.
In this section, multi-disciplinary optimisation processes will be explained, starting with information about optimisation and how disciplinary optimisation might interfere with the optimal solution of the whole. Then design problems will be explored to find out what kind of information is needed to have the team cooperate. As last the process of a multi-disciplinary optimisation will be explored if this has to be set up for a diverse team.
2.1 Optimisation

The optimisation of the design is originally carried out with respect to one governing objective, for example, cost, weight, sustainability, buildability, etc. But what is optimisation of the design and how should this be done? Optimisation problems have always been intriguing and have been around for a long time, probably one of the earliest structural optimisation problems was Galileo Galilei’s search for the strongest cantilever beam in bending and constant shear. Here there should be optimised for minimum weight under a uniform stress constraint, causing the beam to be strong enough to withstand the forces but light enough, not to break due to self-weight.

A typical building design involves a wide range of disparate disciplines – architecture, structure, building services, landscape designers, master planners, fire engineers, etc., working together for a relatively short period on the design of a building [6]. All or at least most of these disciplines are trained and educated discipline-wise, meaning they have little or insufficient knowledge about the other domains and their tacit knowledge.

From a mathematical point of view optimisation is done via a function which represents the measure of performance, and independent variables spanning the design space [7]. The optimisation to only one parameter (function) mostly happened because traditionally engineering problems tend to be solved sequentially. Separate teams are specified for their explicit discipline; all obtaining the pre-set preconditions (design space). Because of the parallel processing they are not able to change or respond to new gained insight from the other parties, but are only performance driven to their own discipline. However, when the design process is completed, it is evaluated with respect to its performance in all of the areas.

A first step into the right direction came with the introduction of computer-aided design, which allowed designers to quickly modify and analyse their designs. This made it possible to react upon the other parties without extensive extra analysis, enabling the creative designer to supplement intuition with computational tools in order to verify the validity of the new concepts. The next step is to have collaborating teams with experts from multiple disciplines working on the same project. These cross-functional teams are mandatory to develop multi-disciplinary products. The focus could then shift towards more economic driven design, leading to optimising the criteria (mostly cost, or aesthetics) with all disciplines involved.

[6] Z. Ren et al., 2011
2.2 Interactive optimisation

The complex nature of building design characterises its design queries as multi-parameter, multi-discipline and multi-objective. Therefore when building designs become more complex and decentralised, it requires effort from multi-disciplinary design teams. Multi-disciplinary design optimisation allows the design teams to incorporate all the relevant disciplines simultaneously considering the interaction between the disciplines. This way they can find an optimum of all disciplines together instead of sequentially optimised disciplines added together. Although this leads to a better optimum of the problem it does introduce a higher complexity to the solution. Because of their multiple parameters, multi-disciplinary optimisation problems are non-linear optimisation problems, most of the time containing multiple optimums.

The interaction between different disciplines is the key factor here, but is it possible to quantify the influence or importance of each discipline separately? A good ‘whole’ design cannot be achieved by concentrating on the ‘parts’, since each part may pull the project in a direction that might seem regressive to the other ‘parts’ \(^8\). Where an optimum solution with all constraints set to their combined maximum might lead to very low values compared to their individual maximum. This is because this approach focuses on maximizing the joint probability of the objective with all constraints being satisfied, instead of individual optimums.

2.3 Design problems

This brings us to the complexity of design problems. Design is about solving problems, either externally or internally driven, in the best possible, most beautiful or elegant way. Elegance in designing has to do with choosing your parameters intelligently, then optimising the solution of a specific problem and to do so whilst holding the criteria constant. You will never be able to show that you have achieved the highest possible quantifiable score on every criterion simultaneously. Architectural problems tend to be both over and under determined; no solution will satisfy all constraints to the desired degree, but many solutions will provide similar levels of satisfaction. \(^9\)

It is satisfaction that makes optimising a qualitative instead of a quantifiable criterion. For people to be able to make qualitative decisions it is important to have similar levels of understanding of the matter and the same type of presentation of the data. This will be addressed into more detail later on in the thesis.

\(^{[8]}\) Z. Ren 2011
\(^{[9]}\) Uijtenhaak 2010
2.4 Setting up a Multi-Disciplinary Optimisation project

Before going into detail about finding the most optimal outcome via Multi-Disciplinary Optimisation (MDO) it would be good to focus on what MDO is and the way a MDO project could be set up. MDO is a method used for the optimisation of large and complex engineering designs. Most of these designs exist of complex interrelated systems in which the synergistic effects of coupling between various interacting disciplines are exploited at every stage of the design process. MDO models the interaction between the different components of the problem. It is based on a decomposition principle subdividing the main problem into several sub-problems. The design problem is then tackled by specialist who start analysing the inter-relationships, represented by variables, between the different sub-problems. The multidisciplinary nature of most design problems complicates model choice and implementation. Often several iterative steps are necessary between the disciplines in order to find the values of the objectives and constraints.

2.4.1 Decomposition

A first step into MDO is decomposition of the tasks at hand into individual elements. Thereby streamlining inter-disciplinary communication and finding parallels between analysis and optimisation by coarse-grained methods. In this way a first indication of the tight coupling between optimisation and analysis could efficiently be improved. The structure of the problem could be simplified and some of the many design variables (a specification that is controllable from the point of view of the designer) involving the multidisciplinary analysis disciplines could be mapped, something particularly difficult in design problems with a large design space and loose boundaries. Doing so, many specialists are working in various disciplines; all being dependent on each other, know some of the complexity and the dependency off the other groups.

For a project to work well a chief designer or core team, dealing with the design tools and communication between the different organisations, should be appointed in the start meeting by the whole team. This person or team can then keep an eye on the process and the collaboration between the teams and take the lead when deadlines are approaching.

2.4.2 Organisation

As design problems become more complex, the role of disciplinary specialist increases and it becomes more difficult for a central group to manage the process. As the analysis and design tasks become more decentralised, the
importance of a sound internal communication system where all information is stored and can be traced is of vital importance. All constraints, design variables, the interdisciplinary variables and appointments made with the client should be documented well and stored in a place for all parties to be easily accessible. This should be one of the tasks of the chief-designer or conductor. This document should clearly outline what should be considered among different disciplines and what will only be considered within each individual discipline, along with the dependences and impacts between each discipline.

The steps considered here include, the simplification and decomposition of analyses using numerical optimisation, and the transformation of the design problem itself into parallel, collaborative tasks for the different disciplines. Thereby defining three types of variables, local (only for the single separated disciplines), shared (more than one discipline) and coupling (output from one discipline, needed for the next)\cite{12}. With the shared variables there are again two possibilities. The relation between the parameters can be because of designation of the entity, or the relations intrinsically share one common concept, in the last case it is not a design choice but pure physics that determines the linkage between the theories \cite{13}.

If these extensive preparations are done and documented, all parties know what to do and when to interact with other involved disciplines. Now the decomposition permits disciplinary problems to be addressed by experts who understand the physical significance of the variable or constraint, and know how best to solve the local problem. In this way of organising the MDO problem, all disciplines will be fully optimised even though the process of developing is still a parallel process.

2.4.3 Parallel processing

The parallel processes can be seen as the traditional way of designing, but in this case there will be a set authority (the chief designer). This authority will check the outcome of the separate disciplines and see if the final solution is within the set boundaries and whether the sub-solutions provided conflicts with other sub-solutions. Of course it should be taken care of that all separate disciplines can function as efficient as possible and within their own design responsibilities. For local groups to succeed it is important to sit together in the course of the design process and sometimes disagree with the other disciplines. In the end it should be the goal to minimise these interdisciplinary disagreements and come up with a satisfying solution that matches the system target \cite{14}.

The decentralised groups should satisfy their local constraints so that no domain
specific information needs to be communicated with the other disciplines. The shared constraints need to be fine-tuned within the responsible disciplines and the coupling variables should be communicated between all the groups.

2.5 Conclusion

The previous section showed clearly how design processes work and how different parties have different concerns. When all disciplines work without interacting the total design would be a mixture of optimal solutions leading to a sub-optimal end solution because there will be no coherency. Therefore for the tool it would be very important to represent the boundaries or overlapping areas with other disciplines. At the same time it shows that decentralising or parallel processing is very important and therefore the dashboard should merely function as discipline specific tool providing in-depth information but at the same time creating awareness of the boundaries caused by the multi-disciplinary character of the problem.
Computer-aided design and drafting systems have proven to be very important tools in the building industry. In traditional CAD systems, geometric modelling is performed by manually drawing the geometry. Further down into the process all dimensions are then derived from the drawn geometry. This means that all parties are required to know the precise dimensions of the geometry at the very beginning of the project. Changes later on are difficult and very time consuming. But variations in design are a fundamental part of the design process, especially in the conceptual phase\textsuperscript{[15]}. 

\textsuperscript{[15]} Carlos Roberto Barrios Hernandez, 2005
Knowledge (tacit) from previous phases or attempts supports improvements in the design which in turn improves the quality of the total design is therefore inevitable. Design is an iterative process of constantly going back and forth between different alternatives which all could be possible solutions. Therefore in the conceptual design phase there is a need for a flexible tool in which the user is not always sure about the final design outcome. The solution to this could be in parametric design, which allows designers to make modifications to existing designs by changing parameter values.

3.1 Parametric and associative design

Parametric design and optimisation are becoming common terms in the current building atmosphere. One of the reasons for this is because of the extra computer capacity and improved computational skills; these are leading to new ideas and new drivers for design. Computers can be used to generate innovative alternatives via changes in the parameters that underlay design. Parametric design is the process of designing in an environment where changes are effortless, replacing singularity with multiplicity in the design process. Parametric design is done with the aid of parametric models, a computer representation of a design constructed with geometrical entities that have properties (or relations) that are fixed and others that can vary. The user-variable attributes are called parameters and the fixed attributes are called constraints. The parametric model can provide a consistency in design logic via the prebuilt rules and relations, this makes it easier to work with different parties involved if you know what can or will change. The basic sophisticated start idea will still carry on even though changes in the parameters are made.

“Detailing in a parametric context is neither a question of scale nor dependent on fixed parameters, but rather depends on design logic and the correct parametric relations.” (Dominique Holzer, 2009)

Parametric design is, in a sense, a rather restricted term; it implies the use of parameters to define a form, when what is actually in play is the use of relations (for example if the floor height changes in a high rise building, this should change on every floor. In a parametric environment all the floors will than adapt and still stay connected). But for now parametric design will be seen mainly as design logic for rules and relations.

Regardless of the performance and complexity, all parametric models can be categorised into two kinds: those that perform variations and those that generate new designs by a combination of parameterised geometrical entities or components.  


[17] Carlos Roberto Barrios Hernandez, 2005
Figure 3.1
Primary set to 3 parts and Secondary set to 4 beams

Figure 3.2
Primary set to 15 parts and Secondary set to 13 beams
3.2 Constraint based

The constraint-based model is a parametric model based on the declarative nature of the parameters to build shapes. The designer creates a geometrical model, and all its attributes are parameterised based on the desired behaviour. A parametric modelling schema shows which attributes of a geometrical model are parameterised and how the designer can change the values of the parameters. The idea behind this type of model is that the geometrical components are controlled by means of changing the values of the parameters or constraints without changing the topology (number of components and their relations). The sequence of operations is stored so that it can be executed again when parameter values are modified. The parametric modelling schema is the starting point for parametric variations in the designs. Every time the designer changes a predefined parameter a different design instance is created.

To provide an example of constrained based parametric models one can think of a hall with a primary and secondary structure. Here the example exists of three base lines that will be connected via portal frames. These portal frames (primary structure) will then be connected via beams (secondary structure). With this type of parametric design it is possible to change the amount of primary and secondary structure by simply changing the constraints. If the designer only wants to have a primary structure, he could change the constraints in such a way that the portals are spaced with only the distance that the roof panels can span. On the other hand if the designer wants to look for as little material as possible he could optimise the structure in such a way that the combination of primary and secondary structure leads to the minimum amount of steel. Constraint based parametric provides the opportunity to go through many alternatives with only changing two constraints. The logic stays the same, but the outcome is different.

3.3 Associative combination

The other type is associative combinations, a model composed of a series of geometrical shapes (associations). These shapes can then be arranged according to rules that create more complex structures out of the prefixed components or modules. These models are also known as associative geometry models or relational models.

Associative combinations offer another degree of complexity beyond the parameterisation of the geometrical components, which is done by constructing combinations according to specific rules. In associative combination models, the important aspect is the spatial relations and rules of combination between the
Figure 3.3
line transformed to a HE100A profile

Figure 3.4
line transformed to a CHS100A profile
primitive components, which determines different design compositions. To provide an example of associative based parametric models one can think of a model in which elements such as beams can be given certain properties, such as an HE-beam or a CHS-beam. This time you assign a property to a line.

3.4 Relation based

Both types work with a set of finite instructions that perform specific tasks. These subroutines or functions work like a procedure which takes the parameters as input and runs them through the predefined logic to compute an answer. Both design procedures offer the designer a powerful way to generate several parametric models which they can then explore further. The two types can also be combined into a parametric associative model\cite{18}. These parametric models are the beginning of a framework for high level manipulation of the geometric components, which transform the outcome and provide real-time feedback of the solutions.

Now it becomes more of interest to communicate the relations (of the model) between all parties involved, rather than only the developers of the parametric design (e.g. the project developer should be able to understand and participate in the decisions made by the architect or engineers, who made the model of the building). With these enormous possibilities, it is important to understand what the changes in parameters do. Much information can be generated through working with these relations, but the data generated is not always useful. Sitting together with all parties involved staring at the parameters will not end up in the wanted design. A scoring should be appointed to every possible alternative, and these should then be displayed. This could be done via a dashboard representing all valid options.

3.5 Conclusion

Parametric design is important for this tool, since as shown in the section about design processes or multi-disciplinary optimisation the tool should be very agile and flexible but at the same time provide a strict structure preventing miscommunication and therefore errors. Parametric design is perfectly suited for a job like this, since it works with a set design logic that can vary by set constraints. The possibility to freely change all parameters within the previously set boarders reduces concatenation errors. The possibility to automate the process of changing all constraints and let the computer crunch through all options will generate the data wanted for further examination.
Visual means are very important to the quantitative aspect of designing, they make it possible for the non-expert people to ‘understand’ what their idea is. Especially with a complicated design query as created by multidisciplinary optimisations. Communicating alternatives, optima or why one favours a solution could best be done visually. The visual representation of different solutions can be provided as a means for capturing and enabling designer insights. This could allow the designer to make decisions before, during or after analysis or optimisation via a visual environment, in order to effectively steer the solution process.
Figure 4.1
BonaVista dashboard, an example of information at a glance, image from bonavistasystems.com
The graphical or visual representation could lead to a better solution in less time by exploiting the knowledge and expertise of the expert. Complex design could then be broken down to tipping points, black swans and interrelations that stand out.

A dashboard displaying the ‘actionable’ information would be highly suitable for situations as shown in figure 4.1.

### 4.1 Dashboard Layout

Dashboards are excellent tools for summarizing data on one single screen (website) and allowing users to comprehend what is otherwise information overload.

> A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.” (Stephen Few 2004)

Dashboards can display real-time information with only the most important graphs or gauges and some additional text. Besides real-time information they are extremely suited to compare information, dates from the past compared with current values, or just a comparison between multiple alternatives. They can provide the business intelligence without digging deep into the data and interrupting the workflow performing bidirectional communication with the data-sources.

The efficient way of presenting information can then be used to gain insight in the matter. Problem areas and opportunities are highlighted encouraging the user to take action; this is the primary job of the dashboard. This action could be taken by either clicking on the graphs for drilling down and entering new levels, or navigating to the underlying information within the system itself, but now you know where and what to look for.

With MDO designs set up in a parametric way, a lot of data that is generated could be efficiently displayed on the dashboard. But before the data can be displayed it should acquire some value. Data should therefore be organised and once that is done the data can be seen as information. This organising of the data can be done via several modules connected to the parametric model. All data will then be stored in a database, which will function as the organised structure that the dashboard has to graphically display. The displayed data on the dashboard can then become knowledge for the people behind the display. Therefore all this data is of vital importance for the success of the process and should be carefully stored in a database.
Figure 4.2
A scheme showing how the dashboard connects its users to the database, which has all the information of the design model stored.
In this thesis the dashboard is focused on the structural engineer and how he/she can benefit from this. Therefore it is necessary to provide the dashboard with different sliders/operators/filters via which the engineer can browse true all the data stored in the database. Presenting all data visually makes it possible to screen all options, but by comparing it to the other alternatives as well as have a visual image of how the structure can look like. So the dashboard will help people making discussions based on smart data mining and visual preference.

4.2 Conclusion

The strength of a dashboard to output all the data at a glance is very helpful for the proposed tool, since here you want to have multiple users to quickly understand where the focus areas should be. This will then lead to better substantiated decisions where the relation between choice and consequence is much clearer. The possibility to dive deeper into the data and extensively explore black swans in the graphs will be very helpful to show other parties where the highest profit can be expected.
The largest part of the decision making via dashboards will be done with the use of data mining. To do this sophisticated it is very important to store the data in such a fashion that it can easily be presented again. The Business intelligence vendors have concentrated highly on developing sound underlying technologies for data mining. This technology helps to gather data from source systems transform the data into a more useable form, storing it in high-performance easy accessible databases and present the data in the form of reports\textsuperscript{[19]}. Over time different systems for databases have been invented and these have evolved into smart systems that all deal with the data in their own way.

\textsuperscript{[19]} Stephen Few 2006
## Flat Database

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Name</th>
<th>Length</th>
<th>Amount</th>
<th>Stress</th>
<th>Reaction</th>
<th>Crosses</th>
<th>Angle</th>
<th>Deflection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>1.3</td>
<td>1.4</td>
<td>1.5</td>
<td>1.6</td>
<td>1.7</td>
<td>1.9</td>
<td>1.10</td>
<td>1.11</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
<td>2.2</td>
<td>2.4</td>
<td>2.5</td>
<td>2.6</td>
<td>2.7</td>
<td>2.9</td>
<td>2.10</td>
<td>2.11</td>
</tr>
<tr>
<td>3</td>
<td>3.1</td>
<td>3.2</td>
<td>3.4</td>
<td>3.5</td>
<td>3.6</td>
<td>3.7</td>
<td>3.9</td>
<td>3.10</td>
<td>3.11</td>
</tr>
<tr>
<td>4</td>
<td>4.1</td>
<td>4.2</td>
<td>4.4</td>
<td>4.5</td>
<td>4.6</td>
<td>4.7</td>
<td>4.9</td>
<td>4.10</td>
<td>4.11</td>
</tr>
<tr>
<td>5</td>
<td>5.1</td>
<td>5.2</td>
<td>5.4</td>
<td>5.5</td>
<td>5.6</td>
<td>5.7</td>
<td>5.9</td>
<td>5.10</td>
<td>5.11</td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
<td>6.2</td>
<td>6.4</td>
<td>6.5</td>
<td>6.6</td>
<td>6.7</td>
<td>6.9</td>
<td>6.10</td>
<td>6.11</td>
</tr>
<tr>
<td>7</td>
<td>7.1</td>
<td>7.2</td>
<td>7.4</td>
<td>7.5</td>
<td>7.6</td>
<td>7.7</td>
<td>7.9</td>
<td>7.10</td>
<td>7.11</td>
</tr>
<tr>
<td>8</td>
<td>8.1</td>
<td>8.2</td>
<td>8.4</td>
<td>8.5</td>
<td>8.6</td>
<td>8.7</td>
<td>8.9</td>
<td>8.10</td>
<td>8.11</td>
</tr>
<tr>
<td>9</td>
<td>9.1</td>
<td>9.2</td>
<td>9.4</td>
<td>9.5</td>
<td>9.6</td>
<td>9.7</td>
<td>9.9</td>
<td>9.10</td>
<td>9.11</td>
</tr>
<tr>
<td>10</td>
<td>10.1</td>
<td>10.2</td>
<td>10.4</td>
<td>10.5</td>
<td>10.6</td>
<td>10.7</td>
<td>10.9</td>
<td>10.10</td>
<td>10.11</td>
</tr>
</tbody>
</table>

Figure 5.1
a screenshot of a Flat Database Structure

Figure 5.2
a screenshot of a Relational Database Structure
5.1 Database structure

Databases are used in many applications; most current computer software is based on such technology (f.i. electronic mail programs and personal organisers). A database is a collection of data, stored on a central computer/network or in a data centre, organised for easy access and management of the data. This is especially convenient for large multi-user applications, where coordination between many users is needed. To access information from the database you will need a database management system, existing of a front- and a back-end. The front-end (the clients) is mainly concerned with data entry, enquiry and reporting. The back-end (the server) is the set of programs that actually control data storage, responding to requests from the front-end. Searching and sorting is usually performed by the server.

Databases can be organised according to their organisational approach, the flat database, the relational database, a network database or an object-oriented database [20].

Flat database
Traditional ('flat') databases have been dominant for a long period of time. These are organised by fields, records, and files. A field is a single piece of information; a record is one complete set of fields; and a file is a collection of records. This can be seen as a spread sheet, limited by a simple row and column structure.

Relational model
With the rapid advance of the computers in the second half of the twentieth century, the development of databases structures changed. The database structure became far more efficient for dealing with large volumes of information. The most notable is the relational model proposed by E.F. Codd in 1970[21]. Codd criticized existing data models for their inability to distinguish between the abstract descriptions of data structures and descriptions of the physical access mechanisms. Today’s relational databases allow users to access, update, and search information based on the relationship of data stored in different interrelated tables. On relational databases one can run queries that involve multiple databases. While early databases could only store text or numeric data, modern databases also let users store other data types such as sound, pictures, and videos.
Figure 5.3
a screenshot of an Object Oriented Database Structure

Figure 5.4
a screenshot of a Network Database Structure
Object oriented database
During the 1990s, object-oriented databases became a new focus of research, to a certain extent because of the great success that the object-oriented concept was having in programming languages (e.g., C#, Python, C++, Java, etc.). The object oriented database contains the same type of objects to represents its information. An object database stores complex data and relationships between data directly, without mapping to relational rows and columns, and this makes them suitable for applications dealing with very complex data. Objects have a many to many relationship and are accessed by the use of pointers. Pointers are linked to objects to establish relationships. Such databases have had some success when it is necessary to deal with extensive and more complex data than relational systems can easily cope with. Examples could be multimedia and engineering data. But this is not a very common type of database.

Network model
The network model allows multiple datasets to be used together through the use of references (or pointers). Some columns contain pointers to different tables instead of data. Thus, the tables are related by references, which can be viewed as a network structure.

5.2 Conclusion
Databases provide the possibility to access the data without having all the expensive software, to generate and calculate data, installed. If the databases are accessible via the internet the advantages are even higher since then multiple disciplines spread over different offices can access it and professional databases make sure the data is stored save and there is next to no downtime. The easy accessibility of the data via the advanced database technology provides the option to store all, brute force generated, data generated and data mine the solutions quickly via the dashboard.
PART B

TOOL DEVELOPMENT
Design (in this case architectural) is a matter of taste, beside showing that you understand the brief of the project developer, everything comes down to convincing that your plan will suit his wishes the best. Therefore the design should not only satisfy the wishes but also fill the client with pride. As shown before, design is an ill structured process, this leads to a situation where the existing and the desired state are unclear and, hence methods of reaching the desired state cannot easily be found. Everyone reads the brief in a different way and all selected parties will come up with a different design. It will not be possible to quantify all possibilities and then rationally select a winner of the competition.
Looking back at the design process it is all about setting up a (conceptual) design. This can be seen as the first four phases as stated in the conceptual/preliminary design phase, about gaining knowledge of the design question and showing precedents, extended with the start of designing. In a competition this will be done by several different parties, who are selected for the job. After these phases they will present their outcome to the client or initiator of the design assignment and it is up to him to decide who will win the completion and is selected for the final design. Therefore the decision phase is a very important moment for the design team. This is the moment where they should show the quality of their work (design) and impress via a sound presentation with a lot of beautiful visual material.

Presenting and convincing that what you have done is of very high standard brings us to one of the problems in multi-disciplinary optimisation. Not all disciplines have the same skill, or same high quality tools for presenting their added value. Some information is more visual than others (f.i. architecture and landscaping are the main eye-catchers), and most of the time this is governing in the decision. To convince the client, who most of the time has little in-depth knowledge, it is very important to arouse some emotion and provide him with the feeling that the team has thought of everything and takes away all his concerns. An example could be:

- Buildings by Santiago Calatrava, his designs are not always structurally optimal, but the exaggeration and artistic freedom with which he strengthens and highlights certain elements of his design creates a highly elegant image that arouses the people who see the buildings;

- Buildings by Ludwig Mies von der Rohe, an architect who was very highly skilled in the tectonics of architecture, creating highly sophisticated and extremely high detailed work which shows his passion for design and transfers that to anyone who looks at his buildings.
Figure 6.2
TGV train station in Liège Guillemin by Santiago Calatrava

Figure 6.3 a & b
Seagram building
New York, by Ludwig Mies van der Rohe
The same goes for within the team, the most visually advanced part of the team will be the leading force of the design, since they can persuade the others to opt for their solution. If the emotions raised by the disciplines are not of the same standard, the solution will shift towards the most visually attractive ‘short term’ solution instead of the solution that might end up with a far better result on the long run.

Precedents are very important to show what the solution can look like, what to watch out for or why not to go down this path. Humans have always built according to examples or precedents. According to Vitruvius, famous for his book De Architectura, architecture is an imitation of nature. He believed that houses originated from nature, saying:

“...Therefore it was the discovery of fire that originally gave rise to the coming together of men, to the deliberative assembly, and to social intercourse. And so, as they kept coming together in greater numbers into one place, [...] they began in that first assembly to construct shelters. Some made them of green boughs, others dug caves on mountain sides, and some, in imitation of the nests of swallows and the way they built, made places of refuge out of mud and twigs. Next, by observing the shelters of others and adding new detail to their own inceptions, they constructed better and better kinds of huts as time went on.” (Vitruvius, 1914 :62)

But not only precedents can provide visual images, nowadays computers can generate near-perfect images of how the design would look like in reality, this is a very helpful tool and very powerful in decision-making. But engineering tools are as yet not very accessible.

When for instance a project developer looks at these images above he might not get very enthusiastic about the second, third and fourth ones, nor will this information help him improve his decision making. As a result the influence of the engineer or other experts might get devaluated and this can lead to projects that will have a less multi-disciplinary character. Where on the other hand the near real image of the architect is something he can get excited by and show to friends and family showing them and telling them about the wonderful project he is going to realise. This leads to the situation that the engineer has become ‘the unseen hand’ in the design, where Starchitects are the engineers’ counterparts. The problem seems to arise from the fact that once the job is not easily communicated between different parties concerned, it is hard to understand its value. Therefore in this thesis the aim is to come up with a ‘decision making’ tool which can communicate the influences of multidisciplinary aspects in a way that everyone can understand its value. Once for instance the project developer can show what kind of choices he has made to his/her peers, this will bring the role of the experts more to the forefront and lead to a better mutual understanding all round. The better understanding leads to more knowledge, and this will influence our choices, since what we do is based on what we know.
Figure 6.4
Architect render of a design proposal for an Olympic train station in Amsterdam 2028, Tijl Uijtenhaak 2010.

Figure 6.5
A piping diagram of how the layout of pipes in a building could be, copyright Groupe Canam.
Figure 6.6
Engineers solutions to a simply supported beam.

Figure 6.7
Engineers solutions to a truss and the stresses in the chords and members with a Unified distributed load.
The aim of this thesis is a ‘decision supporting’ tool, a tool that helps visualising the relation between choices and consequences and therefore enables the user to make a deliberate decision. The tool should help in the process of multi-disciplinary design optimisation, via increasing mutual understanding, or ‘sharing’ tacit knowledge through a visual environment. Therefore it should be possible to change the (discipline individual) criteria and work towards an ‘optimal’ whole solution. But an optimum is hard to define, especially if it contains Multi-Disciplinary Objects (such as sustainability, construction, daylight, etc.).
Figure 7.1
Different parties involved for the façade.
7.1 Problem

The outcome can be easily manipulated by changing the importance of certain criteria. Some problems can easily be optimised, but generally these are problems where few parameters/relations are involved. Once the problems get more complex, more relations will appear, meaning that the optimisation process becomes less straightforward. To gain insight in the interaction between the disciplines this should be displayed on a dashboard and provide real-time information about the status quo. This will help understanding the influence of all different disciplines when they are presented in a comparable way, and in a similar hierarchy.

In figure 7.1 a scheme is provided which should show the complexity of multi-disciplinary design in a building element, in this case designing a façade. The façade is an important element of the building; it provides light and therefore warmth, or shade to the work or living areas behind it. The façade can be used for ventilation and determines the look of a building. Higher floors, means a more comfortable space, more façade, more daylight, less energy for artificial light, more energy for cooling in the summer, maybe some space for solar cells, impressive building, etc. But the façade is one of the most expensive elements of the building and therefore is an element that should be taken great care of. A decision support tool can provide more options and therefore generate ‘self-made precedents’ of the design at hand. This will enlarge the understanding of the influences of the parameters and by growing comprehension it will be easier to push the edges and achieve more quality for the same price.

In multi-parameter problems, parametric design can be of great help, because if the rational model is considered, the design process will gain information during the process, but the steps are predictable. All this information can then be translated to relations in the parametric model generating data, stored in databases. The parametric model will act as the backbone of the design, and can be extended with different modules all providing extra data to the database. The data can be presented on a dashboard, showing where improvement is possible, or where decisions have a positive or negative effect.
7.2 Research objectives

From the problem description above a research objective and sub objectives can be derived. The main research objective is:

**research objective**
The aim is to design a program with which parametric design can be evaluated (by an expert), and even on a multidisciplinary level can be used to reach the ‘most desirable’ solution, while having an understandable range of solutions in which the criteria and relations guide one’s gut feeling in selecting the outcome. This range will then bring common tendencies of understanding of the criteria to light, and the effects the relations have concerning the outcome.

1st sub objective
An important sub objective is that:
Develop a dashboard via which an expert can see the influences of parameters and relations that are bound to the criteria they change. The criteria should hold a multi-disciplinary design approach but this should not lead to a largely extended dashboard with separate fields of interest for different parties.

2nd sub objective
Another sub objective is:
The dashboard should have a generative design in which alterations in the operators are easily made possible, and generate a new solutions frame.

3rd sub objective
Another sub objective is:
The dashboard should be a program that can run without any preinstalled software and therefore it might be wise to have this program web-based. There should be looked into the possibilities of having a secured webpage in which the project can be seen and analysed and even some comments can be left for changing criteria.
The next part will focus on how a tool could be built that satisfies the previously defined research objective and become a useful tool for MDO processes. Showing the discipline using the dashboard how their choices affect the total design.
Figure 8.1
Design scheme
8.1 Approach

It is in the forming of a concept that computers can be of great help. Once the restraints are explored and the first ideas are created, a computer can easily generate alternatives based on the presumptions/relations that have been set. This doesn’t necessarily lead to a concrete concept, but still allows for freedom in changing the parametric model should an important insight come to light. It is also possible to use a computer for certain aspects of an initial idea, and to later combine these in the design by hand.

The justifying of the concept is also a process of great interest, and a computer can be of help. Once the geometry of many conceptual alternatives has been generated (1), these can be analysed with different add-on modules. After that a selection should be made of the most favourable ones (2). Therefore an algorithm should be written via which all the alternatives are tested. The next step would be an algorithm that can then bring to light those options that satisfy the design criteria. The displaying of these options can be a big advantage (3), once you can still tweak the boundaries that are set, and maybe even provide the program with one’s preference in outcome.

Breaking it down into steps, the tool will consist of three separate phase elements

1. Generating the alternatives
2. Browsing the alternatives
A few important lessons can be learned from the previous sections about design. How do all disciplines work together and what should a computational tool do to improve this process. This is important for now but maybe even more for how this could be a helpful tool in the future. Therefore the next section will focus on the steps that need to be taken to build such a tool. Thereby an outlook will be presented on how this tool can evolve in the future to become a modular generic tool of which multidisciplinary projects will benefit.
Figure 9.1
a schematisation of the workflow process divided over two phases.
The scheme to the tool stays the same as the scheme showed in Figure 9.1. The figure is the visualisation of the described approach from chapter 9. All 3 phases will extensively be explained in the next sections. Before the 3 phases will be explained first the workflow will be presented and then the framework will be elaborated.

9.1 Workflow

The workflow (figure 9.1) of this tool is based on a ‘two-phase’ schedule.

Phase 1 could be:

1. First a sketch design has to be made and the changing parameters governing the design have to be chosen; once this is done it can then be parametrically set up;

2. Once the structural layout is derived and the structural concept is implemented, the governing loads need to be added. This can then be redirected to analysis software analysing the model and adding the outcome to the database;

3. Now all parameters can be controlled by code that changes all the parameters and stores all possibilities in the database. This can also be seen as the first part of the figure 9.1 scheme.

The 2nd phase could then be:

1. All this data from the database has to be implemented on the dashboard and can then be addressed to support decision making by the engineer (the second part of the scheme);

2. Now all options are analysed the thing that has to be explored is the relations between the parameters and the effects of certain design choices (the third part of the scheme).

Dependent on the personal preference and discipline different styles of dashboards can be used, either very visual or numeric. The dashboard is set up in such a fashion that it functions as a coarse grained filter of the data. Every step the decisions made will lead to more insight in the data. This will all run on a framework, to provide a modular and generic structure. The framework will be covered next.
Figure 9.2
A scheme of the process from model to Result.
9.2 Coding the framework

The idea is to build a modular tool, in which new modules can be added or changed. All modules will work separately but need to be joined on a framework. This framework will function as the main tool translating the separate pieces into useful data and pushing it from on level to the next. The coding language for this structure will be Python.

9.3 Python code

Python is a powerful interpreted programming language, using indentation for block delimiters. It has an effective approach to object oriented programming and provides high-level data structures such as list and associative arrays (called dictionaries), dynamic typing and dynamic binding, modules, classes, exceptions, etc.\(^\text{[22]}\). This is of course nothing spectacular and most of the object oriented programming languages have these qualities. But Python is remarkably elegant, and not overly cryptic. Its syntax emphasizes support for common programming methodology and promotes code readability and thus maintainability. It combines remarkable power with very clear syntax. Besides that an important goal of the Python developers is making Python fun to use. This is reflected in the origin of the name, based on the television series Monty Python’s Flying Circus\(^\text{[23]}\).

In the research, Python is used as a high level language and the core of the framework. The large standard library, commonly cited as one of Python’s greatest strengths, provides pre-written tools suited to many tasks. Rather than writing all the components, Python is used to extend the existing programs with modules or components implementing specific functionality. The high-level language serves as a ‘glue’ to tie modules and components together and to rapidly create specialized applications.

The scheme above shows the influence of python as glue between the components. Every step is done with the most suitable program and then via python linked to the other components in the framework.

9.4 Generating the alternatives

This part is about the generating of the alternatives, this all happens after the brief is understood and the first sketches are made. Then the geometry can be set up to really get the process going.
9.4.1. Geometry

Parametric modelling

In the first part of the design process parametric model will be made. This should be done after the design team has been brought together and the first ideas are sketched. In this thesis the tool that will be used for parametric modelling is Grasshopper, a plugin to Rhino3D.

Rhinoceros

Rhinoceros (Rhinoceros) is a stand-alone, commercial NURBS-based 3-D modelling tool, developed by Robert McNeel & Associates. This CAD program is a commonly used program in the building industry; many architectural firms make use of this software to model their designs. Previously in Rhino4, it could only be operated via a .Net language combined with RhinoCommon, but since Rhino5 (which is at this moment still a Work In Progress version, as a Beta release) Python can be used to control and run Rhino.

In this thesis Rhino5 will be used to set up the parametric model, but this is mainly done with a plugin to Rhino, called grasshopper.

Grasshopper

Grasshopper™ is a visual programming language developed by David Rutten at Robert McNeel & Associates. Grasshopper runs within the Rhinoceros 3D CAD application and is a visual programming language. Grasshopper consists of pre-programmed components that can be dragged onto a design canvas. These components can then be connected via wires. Visually this looks like an electrical circuit board with all the elements on it. Via the interconnections of the components certain design logic is set out. The logic will stay intact when you alter the parameters. This provides the option to create several alternatives and visualise these in the Rhino window. The ‘models’ in Rhino can then be rendered or showed to the different parties. Whenever changes are made, but the logic stays the same, the new solution could be generated within ample seconds.

Later on the dashboard will still show these parameters, since this is still an important part for the decision making. The parameters from the geometry can
still be seen and influenced from the dashboard, functioning as the first ‘filter’ of the dashboard.

Figure 9.5 shows multiple options for a set volume, here the parameters could be the width depth, height of the building block and whether or not an atrium (and width and depth of atrium), etc. This leads to different type of blocks, from flat and fat to high and slender. If the restrictions can be set in the first part of the dashboard then the options will reduce severely, speeding up the analysing process. This will work as the first set of choices, which could be seen as a decision tree.

9.4.2. Analysis

From the above it can be seen that the combination of Rhino and Grasshopper proved to be valuable tools for representing the design. A visualisation of the design can be made and this can be used to communicate with the different disciplines. But this is only the surface of what the design should do. The aim of this thesis is to combine all disciplines, have them collaborate and investigate how optimisation of the total can be reached by digging into the details. Therefore several analyses should be made from the outline of the design.

“Engineering is the art of the possible; analysis is the science to investigate the details.” (oasys-software.com)

Different analysis software can be coupled to the Rhino model and the geometry can be exported for extensive calculations. For now the focus will be on the structural part of the design, but many more modules (such as Radiance, DIAlux, Energy plus, Beams etc.) can be programmed and added to this. This would make it possible to upgrade the whole process to a multi-disciplinary project. In the next section it will be explained how GSA, a structural analysis program, can be implemented in the parametric model.

**GSA**

GSA is structural analysis software designed by Oasys, the software house of Arup, and was built by ‘structural’ engineers for ‘structural’ engineers. GSA enables you to analyse and design a range of structural models composed of skeletal frames and two-dimensional finite elements, which makes it the perfect tool for the ever increasing demands placed upon structural engineers. Creating the model in Rhino and then building the model again into GSA would be very laborious, therefore Arup (Paul Jeffries) build Salamander, a tool to convert Rhino models to GSA models.
Figure 9.4
'Visualised' same volume and FSI but a different GSI

Figure 9.5
Decision tree for the volume options, showing the speed of narrowing the options.
**Salamander**
The Rhino ‘Architectural’ geometry should be converted to structural geometry or so called Salamander objects, which can then be exported to GSA objects. This can be done in the main Rhino window, or to make the program parametric there are special components that can be added to Grasshopper. Salamander changes the lines and points from Grasshopper, representing the structure, into 1D element such as beams and changes points into nodes. To these 1d elements properties can be assigned, such as profiles, type, dimension and material. The nodes, which are the connections between the 1d elements, can be set to restraints, such as hinges, or moment resistance. These properties are the same properties as will be applied in GSA. Here it is very important to know which structural concept will be applied, since all these steps will be added to the design logic and will not change if the parameters change. When these steps have been taken, the loads can be applied, via special salamander components in grasshopper, to the structure and the module can be exported to GSA where it can be analysed.

**Other programs/modules**
Likewise Salamander, other programs or plug-ins can be designed to fit other needs. The open source quality of grasshopper allows programmers to extend and exploit the possibilities. Interoperable software can bridge the gap between layout and analysis. It would for instance be very useful to have a software program that outputs a detailed wind analysis. In the future these wind analyses can be done through Computational fluid dynamics (CFD) simulations, these can than provide more accurate information (data) about the actual (lateral) loads on the structure and via this information validate the first assumptions. The extra data from these programs can then be added to the database and be later on used to check if the solution at hand would be the best solution. Another aspect that is of great importance is the amount of light in the building; this can be daylight or artificial light. Both types of light determine the radiation and heating of the building which is important for the amount of ventilation.

A program that could be coupled to Rhino and would be very useful for a daylight analysis would be Radiance, this can provide extra insights in the radiation from the sun. On basis of this information, the orientation of the building, the height of the floors, the percentage of glass and the sun shading can all be calculated and defined. Other programs that would be useful to add to the existing modules would be a program such as DIAlux, a program that is used by lighting designers to see how much artificial light is needed for a perfect illuminated environment. All this
**MDO - Decision Database**

<table>
<thead>
<tr>
<th>Data-ID</th>
<th>Data-Col1</th>
<th>Data-Col2</th>
<th>Data-Col3</th>
<th>Data-Col4</th>
<th>Data-Col5</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>adipiscing</td>
<td>410673</td>
<td>35°</td>
<td>157 kN</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>turpis</td>
<td>310629</td>
<td>15°</td>
<td>67 kN</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>vehicula</td>
<td>49099</td>
<td>31°</td>
<td>2896 kN</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>sagittis</td>
<td>71073</td>
<td>75°</td>
<td>16 kN</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>scelerisque</td>
<td>410674</td>
<td>3.4&quot;</td>
<td>2346 kN</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>augue</td>
<td>406510</td>
<td>46°</td>
<td>1500 kN</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>venenatis</td>
<td>9105703</td>
<td>17°</td>
<td>97 kN</td>
<td></td>
</tr>
</tbody>
</table>

Figure 9.6
*Visualised* database content
data can again be added to the database and be used in the decision making. Basically the options are endless, with the use of Python as 'glue' all programs could be used to provide more detailed information about the model and be stored in the database to use in a real-time analysis of the whole design. The list of possibilities would not be discussed in this thesis, since the aim is only to provide a modular framework that can be extended in many ways to suit the wishes of the design groups.

Again to implement this on the dashboard is important for the decision making. The types of analysis can still be influenced from the dashboard, extra analysis software can be added or some can be removed. At the same time via the settings it should be possible to change the type of analysis and the output variables.

9.4.3. Storing Alternatives

The storage of data is very important in the concept of this thesis. When data is structured in such a fashion that it becomes accessible, it becomes information. Then the next step is to use this information in an organised way and present it so it becomes knowledge. It is tacit knowledge transfer that is the final goal, because when all disciplines can share their knowledge with each other, mutual understanding is one step closer. This mutual understanding should provide more equal information than when all disciplines are having their own knowledge and ways of visualising/presenting and therefore different levels of conviction.

Python is used as the glue of the project and a sort of virtual project manager, but the basis of it all is the database. All data from every module/program will be stored in a database, which can be stored online, so that all parties can access it. This accessing should not be to the unstructured data, but via a dashboard portal, which structures the data and presents it in a visual environment.

9.5 Browsing alternatives

A system containing widgets is introduced to have the individual user decide which data he would like to look for and how he wants this to be displayed. The widget system is set up in a modular system any type of widget can be added later on. The User is free to choose which and how many widgets he wants and thereby how he wants to visualise the data on the dashboard.

The data is ordered and stored in a database (back-end) it should be structured in such a way that a program (front-end) can browse through the options. Using a database in a large datacentre prevents that the parties that access the database need special software and store all data on their network. Web based software is introduced to grant access to this centrally stored data.
9.5.1. Web based software

To build a dashboard or portal via which people can access the data from all over the world a Web development framework should be set up. To build and maintain high-quality web applications there is chosen to use Django\(^\text{[24]}\) as framework. Django, a web framework, provides the programming infrastructure of the applications needed to create a dashboard that represents the information. A web framework works with a MVC structure, which is Model-View Controller:

- Model refers to the data access layer. This layer contains everything about the data; how to access it, how to validate it, which behaviours it has, and the relationships between the data.
- View refers to the part of the system that selects what to display (views) and how to display it (template). The template is the presentation layer and concerns the how something should be displayed on a web page or other type of document. Views is the business logic layer, it contains the logic that can access the model and defers the appropriate templates.
- Controller refers to the part of the system that decides which view to use, depending on user input and handled by the framework itself, accessing the model.

The big advantage of this system is that the components are loosely coupled; each distinct piece has a single key purpose and can be changed individually without affecting the other pieces. Therefore a designer can change the page layout via HTML code, without touching the code that runs the database, where an administrator can work with the database without changing the lay-out\(^\text{[25]}\).

A plugin that could easily be added to the Django Framework and make the internet page run faster and smoother is Dajax, Ajax for Django. Ajax stands for Asynchronous Javascript AndXml. Ajax is a group of technologies of interrelated web development methods used on the client (front) side to create interactive web applications. Via Ajax, web applications can send and receive data from a server asynchronously without interfering with the display and behaviour of the existing page avoiding full page reloads. Ajax makes it possible to keep the part of the website that does not change and only reload the partial changes making the changes appear almost instantaneously.

Dajax is the version of Ajax that is especially designed for Django and functions as an intermediate tool between HTML, CSS and Django using python and almost no lines of JavaScript source code. Using the dajaxice communication core, Dajax implements an abstraction layer between presentation logic managed with JavaScript and your Python business logic\(^\text{[26]}\).
9.5.2. Widgets and Operators

The operators are the most important influential part of the dashboard. They function as the interaction between the data, the parametric model, Django and the output on the dashboard. The optimisation of the project and the success of the dashboard rely on these operators. To prevent the endless amount of operators making it difficult to understand the dashboard again the widget system can be of great help, limiting the amount of operators that need to be displayed.

Widgets
After the Geometry and Analysis parameters are set on the dashboard the widgets can be chosen. The widgets are based on the idea of iGoogle, a webpage which you can personalise to have it effective for your use. Some people might be more visually guided and others might prefer graphs or numerical data. The widgets can be added to the dashboard and these will activate the operators that will be helpful in browsing through the alternatives and optimising the design question at hand. Operators can always be added or removed if they are needed to tweak the results on the dashboard and search for the optimum.

Operators
Where the widgets determine the layout of the dashboard, the operators determine the content of the dashboard. All the data is in the database, the layout of the dashboard chosen, searching through the data and selecting the data you need is the next step. The selecting principle is based on how you would select a mobile phone with provider and contract. You start with choosing a provider, the amount of minutes you want to call, whether you want a touchscreen or not etc. Every decision narrows the possibilities and leads you to the ‘most suitable’ phone. Another example is guessing a famous person someone else has in their mind; via yes/no questions you have to narrow down the options as fast as possible (first question could be if the person is a man (50%), second question Chinese (33%)etc.). You will try to break down the possibilities as fast as possible and then come to the right answer.

In the previous section the decision tree has been showed (figure 9.6), another option is zooming in onto a graph. Each time you zoom in, more detail will be displayed and the differences appear better, making it easier to make your decision. Where in the first graph of figure 10.8, there might be 100.000 options, zooming can lead to 10 options in the graph to the right. This can be achieved by changing the operators in the tool and therefore narrowing down the choices and options.
Figure 9.7
Possible widget parameters and their outlook

Figure 9.8
Filtering from coarse to fine

Min Value  Max Value

Figure 9.9
a possible view of an operator
Data can become information now and information then becomes knowledge. The operators can be seen as the outcomes from the parametric models, their analysis and added fields of interest. They display which variables are chosen to optimise and show the values of the minimum and maximum value derived by the data from the database. If these operators can be set as a range within which all solutions should be exported to the first step, a coarse grained filtering of the solutions is done.

Each time one of these filters is applied a smaller set of ‘results’ are presented in the fashion that is selected via the widgets.

Eventually after all filters are applied, all options displayed on the dashboard satisfy the criteria. Then there are again a few possibilities, one is to activate the widget Export and export the results in an automated report, which could then be handed to the client, or be discussed within the group. Another option is to save the settings but change the dashboard layout and dig deeper into the resulting results. Now new widgets can be selected and this provides the option to look at the chosen set with a different perspective.

Via these steps it is possible to maintain and work with a large set of data and bring this down to a manageable amount of data.

**Ranking data**

The reason why ranking data is important is mainly to compare them. When comparing data and searching for an optimum to compare or score data the same way. This can be done via scalar calculations and measuring the distance between the points. But then the values should have the same signs, and preferably roughly the same size. Therefore a ‘cost-function’ should be determined, a function that scales all the values into comparable values and then calculates the minimum or maximum cost of the total.

If data is not scaled but the values are added for an optimisation several problems can occur:

- Scaling data, if one variable goes from 10 to 1000 and another one from 1 to 1.9 the effect of the second variable will be neglect able and thus not of importance for the optimisation, where it might have a huge impact on the real result;
- Offset, one value starts at 1 and the other at 100, this would mean that the value starting at 1 has no influence at all, and again the impact is quantified wrong according to the importance.

Most of the time it is not possible to score everything as a percentage from the min and max value since it is not known what the maximum or minimum value is. This is one of the advantages of storing all data in a database, now those values are known. In the end the total score should be quantified and this outcome can
then be compared to the other possible solutions. This doesn’t mean that all the data is presented in a percentage of the min and max value, but at least that there is a way to score all alternatives in the design space on the same aspect. When you don’t want to brute force all ‘possible’ solutions another option is to use algorithms for generating new alternatives and optimise their scores. With searching for an optimum it can be very time consuming if there are millions of solutions to calculate and go through, therefore there are several different tactics in writing an algorithm, such as:

- Random searching
- Hill climbing
- Simulated Annealing
- Generic Algorithms
- Etc.

This is a topic that will not be further elaborated, but could be very important in future developments because of increasing the speed of analysis and therefore the possibilities to have real time analysis.

In this thesis there are a few advantages because of brute forcing; one is the possibility of scaling everything in a percentage providing a 1D solution. The second benefit is that providing different filters previously to the optimisation reduces the amount of results a lot. The brute force method would therefore be a very satisfying solution for now.

9.6 Visual interpretation or dashboard

9.6.1. Dashboard scheme

The visual interpretation is the last part of the scheme, but maybe for this thesis the most imperative part. As extensively explained in the previous chapters, visualisation is significant for both experts and non-experts to understand what consequences choices made so far have. For experts the dashboard can show them black swans (sudden drops or rises in a graph) or information they need to zoom into. Other experts could be looking for time schedules, building sequence, photorealistic images of the project etc. For the non–experts the visual representation will make it easier to understand what has been done and a proper visualisation of the process will create emotions with the beholder. If the visualisation is done properly all different disciplines will show up with equal importance and a considered decision can be made with all parties equally judged.
As explained in the previous sections the dashboard is a tool to work through the data in a coarse grained way. All the decisions are made to narrow down the options, making the resulting images far more accessible and better suited for the job. If the above should be schematised is could look something like is suggested in figure 9.10.

The scheme in figure 9.10 shows the green part being the geometry parameters. These determine which parameters will be varied for the database calculations. If these are selected in a sensible way the redundancy can be reduced a lot and the system will work much faster. The red part are the analysis parameters, these are used to select which types of analysis should be done. Again not everything should always be analysed, some disciplines might not be interested in the outcome and this can narrow down the outcome and increase the speed of the system.

As third step the blue part is taken care of, these are the widgets. The widgets determine the way the data is represented on the dashboard and are mainly active after the database has been filled. Some examples could be given, being the rendering of the geometry, or making the movies containing data about the building process or walking through the design. Besides setting the output of the dashboard, they also influence which operators will become active to fine tune the output data on the dashboard. After all these parameters are set it will result in the yellow part. This is the output of the parameters and together with the operators from the widget this design space can be used to search for the ‘optimal’ solution.

9.6.2. Web based software

To make this possible software is needed to program the dashboard and design the layout via the widgets. The visual representation is where in Django the distinction is made between the technical database expert and the designer. This would be in the view section and with a focus on the templates. But this is not only design that is important, providing options and scanning through the database is of vital importance to make the dashboard a success. Therefore extensive research has been done to know what should be elaborated and how this should be presented. Searching through the database can be done via Python and Django in the controller space, but the templates are built in HTML and CSS.

**HTML and CSS**

HTML stands for Hyper Text Mark-up Language and is the dominant language for web pages. A mark-up language is a set of mark-up tags which can be seen as a notation used to annotate a document’s content to give information regarding
Figure 9.10. A schematic representation of filtering data with the dashboard.
the structure of the text or instructions for how it is to be displayed. HTML is written in the form of HTML elements, consisting of tags enclosed in angle brackets. These tags nearly always appear in pairs, a starting and a closing tag. Web browsers use the mark-up languages HTML to interpret and compose visual web pages out of them. At the moment the fourth version of HTML is used. HTML is used as the mark-up language, but to define the appearance or presentation semantics and layout of the HTML page, a different type will be used, namely CSS (Cascading Style Sheet). CSS makes it possible to write separate HTML pages and separate the document content from the presentation. At the moment the third version of CSS is the most common version.

When the program is web based it could be accessible from everywhere, with only a time computer showing the interface, while all the data and calculations are done somewhere else where the software is installed. This would then enable the user to go out and show the client what he has done. Or even have him check different options when on site. Maybe later on it would be possible to use this tool to evaluate buildings that you come across and just verify the solutions you see there as a case study.

9.6.3 Operators

The operators on the dashboard are the interactive intermediate part of the tool. Therefore in the previous section about data mining they were mentioned as well. Their function is to retrieve data from the data base and output this data on the dashboard, guided by the choice made for the widgets. Here the focus is on the outputting quality of the operators. To use the operators in the most efficient way, it is important to know what kind of data the operator is linked to and what the values are. A first selection is made by the widgets guiding the user into a set frame and knowing where to look for. But for the operators the same goes as for the widgets, not all data can be presented the same; therefore operators can have different styles. If you want to provide the main wind-direction you cannot assign this via a min and max value, but you need to provide an angle and direction. Whereas when you want to provide a range in within which the maximum stress should be you might need an operator that functions as a slider with a min and a max value where you can set the borders. Or if you want to assign a system or material you might want to have multiple choice options. See fig 9.12.
Figure 9.11
A flexible tool, possible to use on any machine and everywhere.

Figure 9.12
an example of different styles of operators.

Figure 9.13
a possible outcome of the final dashboard.
Besides the way the operators look and their function, the whole set of operators can function as an operator as well. If hierarchy is implemented then changing of the outputting graphs and narrowing of the solution space can be seen every time an operator gets edited. By moving the operators up and down insight can be gained of the effect of an operator, and whether this is a governing factor. Some might have a big effect whereas other hardly changes the solution space. The same goes for relation between the operators; although different fields are changed the effect might be the same.

Another advantage of brute force generating all solutions, besides having all the min and max values, is that besides the operators as filters, a visual representation of the solution can be given as well. Each time when a solution is derived by changing the parameters a render of this solution is made. These solutions can all be showed onto the dashboard, providing the user with images of the outcome. By browsing through these images, preferences can be assigned, deselecting options that are disliked or saving options that are favoured.

All this data displayed on a dashboard (see fig 10.13) makes it easy to comprehend the data and make well-founded decisions, or at least bring the solution space back to a small list of highly favoured solutions.

Analysing data
When graphs are displayed, there are several things that might be of interest, a few options are given below in figure 9.13. The first graph displays a pivoting point; this shows the point where the negative trend turns into a positive point, searching for an optimal this is an important point to find. The second graph displays a hinge; this is a point where a trend is suddenly changed from direction, a little bit better will suddenly have a much higher price. The third graph displays a black swan; this is a sudden drop in the graph, an example could be the amount of steel and the amount of nodes, at a certain point the nodes might be that close together that they can be used for the façade as well, reducing the extra nodes otherwise needed to connect the façade to the main-structure.

![Graph](image1.png)

**Figure 9.14**
Three important moment in the graphs, (pivoting point, hinge, black swan).
Figure 9.16
A possible outcome of the final dashboard, 5 options selected and compared, in this case the options are selected in the visual database at the bottom and compared via the 3 different types of graphs displaying valuable information to weigh the alternatives.

Figure 9.17
A possible outcome of the final dashboard, four renders showing the different impact of the column spacing.
Figure 9.14 displays multiple lines in a graph, again there are several options, two of them are showed here. Graph 1 shows a region; this can be the region which only satisfies the boundary conditions. The second graph shows multiple values for different cases, each case satisfies other values better, where ever the cross each other, one of them because the better solution. This is an easy way of analysing the results and knowing which system to pick when certain values are reached or governing.

Off course there are many other types of representing the data, such as pie-charts, bar graphs, tables, 3D-scatterplots, etc. All can be used, but for now the focus is on the line-graph.

One of the strengths of a dashboard is that it displays all the data onto one screen. But once the solution space is reduced it would be really nice to compare a few alternatives into more detail. Therefore it would be good if the dashboard could provide an option to go into more depth by clicking the favoured solutions and then change the screen, providing all the data of the selected ones. The next step could be to provide the viewer with a larger detailed image, where the selected option might even be rendered in the location.

Other possibilities than the example given above are possible as well, as is shown in figures 9.16 to 9.20.

Figure 9.15
Two other important types of graphs, (region, change of preference).
Figure 9.18
a possible outcome of the final dashboard, sketched. A film showing how you move through the building, 2D plans of the building and a timeline for the building.
Figure 9.19
A possible outcome of the final dashboard, sketched. A rendered image in the location and graphs and a timeline displaying scores of different alternatives employed in this solution.

Figure 9.20
A possible outcome of the final dashboard, sketched. 4 rendered images in the location, and graphs and textual data displaying scores of the 4 different alternatives.
9.6.4. Workflow of the dashboard

In the previous sections all separate parts of the dashboard are explained, on how they work and with which programs they can be build. The next step is to show how the dashboard could be used. The steps are a logical consequence of the coarse grained filter principle mentioned above as well.

**Step 1**

STAP ONE

In this step the geometrical parameters can be set. These parameters follow from the model previously build in Grasshopper. Whether the program runs on the cloud and should generate real time feedback, or should calculate all options during the night, the speeds could be severely increased if a first filter is applied before running the program and reducing the amount of possibilities to go through. Here you have to provide a minimum value, maximum value and interval to generate the alternatives.

Figure 9.21
Step one of the process.
STEP TWO
In this step the type of Analysis can be chosen. Again all the export options should be implemented in the grasshopper model beforehand, but here there can be chosen, whether you want to activate them.

Figure 9.22
Step two of the process. Selecting Engine 1 (an Analysis program, in a real program the analysis type is named.)
STEP THREE

In this step the widgets that you want can be chosen. The widgets determine the layout of the dashboard and can be personalised, depending on the user’s preference (numerical, graphical, visual, etc.). Extra widgets can be added to change the way of browsing the data and use to dashboard as a decision supporting tool.

Figure 9.23
Step three of the process.
STEP FOUR

This step is to set the widgets and have thereby the output the lay-out of the dashboard. In this case the widget controls the input of the graph as well; here different tabs make it possible to choose the right graph.

Figure 9.24
Step four of the process.
In this step the operators appear after selecting the widget and filling out the parameters in the widgets. These operators can then be changed, influencing the output on the dashboard. Once these are set, the graphs will appear.

Figure 9.25
Step five of the process.
STEP SIX
In step 6 the Graphs will appear and from now on the fine tuning can start.

Figure 9.25
Step six of the process.
STEP SEVEN

Step 7 is a repeating step until you are satisfied. The operators can be changed until the output is analysed and the results provide you an understandable, optimised solution.

Figure 9.26
Step seven of the process.
STEP EIGHT

After all the results are analysed and the chosen parameters are the ones you want to save or discuss, the export widget can be activated. This widget enables you to output the data as a automatically generated report, save your template and settings or store a screenshot image of the dashboard.

9.7 Conclusion

This chapter has showed the outlook of the program, the software used and how it could work. The use of widgets and operators separates the layout from the content and makes it a very agile system, fit for different kind of users.
This part will focus on what can be designed and which parties are involved. This case study will provide insight in how the framework shall work and bring to light were the focus is on. The idea is to set up a modular system that eventually analyses every possible design.
Figure 10.1
Hearst Tower by Foster

Figure 10.2
30 St Mary Axe Tower
by Foster
10.1 Application diagrid tower

In this case study there is chosen to design a high rise tower and see which multi-disciplinary aspects are involved. The structure chosen for the tower is a diagrid structure. The diagrid structural system has been widely used for recent tall buildings due to the structural efficiency and aesthetic potential provided by the unique geometric configuration of the system\cite{27}. The earliest example of a diagrid building is the IBM building in Pittsburgh built in the early 1960s. If the diagrid tower is explored according to the design steps it will clarify what should be taken care of in the parametric design. Later on there will be shown if these variables are local, shared or coupled.

10.2 Architecture

A diagrid is a self-reliant structure; the core of the typical office building has little effect. This makes it possible to have free and clear, unique floor plans. The floor plan is free from columns and structure except for the service core. The service core, when used with a diagrid, needs not carry any load other than some vertical gravity load and can therefore be placed freely. From an architectural point of view this is very convenient since all floor space is usable and the floor space index is very high. This is in favour of one of the 5 points of architecture by Le Corbusier (a very famous Swiss architect/designer 1887-1965), the open floor plan, meaning that the floor space was free to be configured into rooms without concern for supporting walls.

Besides the floor space the angle of the columns in the diagrid determine the look of the building from both the inside and the outside. Changing the angle will change the visual experience of the building. Especially from the inside this can have a huge impact. There are many options for grid sizes, for instance an option could be to have a very large grid with only some very large columns. These large columns can then only be seen from some places inside, but block all the view (A). Another option is having a smaller grid with many smaller columns (B). This option provides a more rhythmic pattern of columns, blocking every window, but only like sun shading (C). All options are possible but need to be explored by the architect to see which he finds the most suitable.

10.3 Structure

The primary structure of a high-rise building can be visualised as a vertical cantilever beam with its base fixed in the ground. The structure has to carry the vertical gravity loads caused by dead and live loads and the lateral wind (and earthquake) loads. These structural loads should be considered with regard to static loads as well as dynamic loads. Lateral loads tend to snap the building or

\cite{27} K. Moon 2009
Figure 10.3
showing the effect of different grid and column sizes

Figure 10.4
wind load on a diagrid

Figure 10.5
load from the floors supported by the diagrid

Figure 10.6
a ‘complex’ diagrid node
topple it\textsuperscript{[28]}, where gravity loads make it fall down. The building must therefore have adequate shear and bending resistance (stiffness) and must not lose its vertical load-carrying capability (strength). For tall buildings with a large height-to-width aspect ratio, stiffness constraints generally govern the design. The maximum deflection is a measurement of the stiffness which should usually be somewhere around a five hundredth of the building height\textsuperscript{[29]}. Two modes of deformation, bending and shear deformation, primarily contribute to the total deformation. The stiffer a structure behaves the lesser deflection or lateral transformation.

In braced tube structures, the diagonal members carry the shear forces and the columns carry the moments through their axial actions. Because the main load carrying structure is on the outside of the building this increases the structural depth and therefore the resistance in bending. In diagrid structures, which use perimeter diagonals and do not have vertical columns, not only the shear but also the bending is carried by the axial forces in the diagonals. This is possible because the diagonal members in diagrid structural systems can carry gravity loads as well as lateral forces due to their triangulated configuration in a distributive and uniform manner. All loads are transferred in continues flow. Except for the triangulation the diagrid should also have horizontal rings, taking care of the buckling of the columns. This combined system of rings and diagonals creates redundancy in the structure and prevents collapse due to a failed portion by redistributing the forces.

The grid of the diagrid can be optimised via changing the angles of the triangle, this can result in 20\% less material\textsuperscript{[30]} than with a conventional system. But there is a drawback, which is constructability. This is because the nodes of diagrids have more elements connecting to a node and are therefore more complicated than those of conventional orthogonal structures and therefore more expensive than those of orthogonal structures. Luckily due to the triangular configuration of the diagrid, rigid connections are not necessary at the nodes, and pin connections using bolts can be made more conveniently at the jobsite.

10.3.1. Calculation

In the next section a hand calculation will be made to get a rough idea how a diagrid works and have a way to check the outcome from the GSA models. In the introduction about a diagrid it was stated that a high-rise building behaves like cantilever beam. Therefore first some formulas will be derived to see how this affects the structure and can be calculated\textsuperscript{[31]}. After that the diagonal-strength or stiffness for lateral forces and bending, will be calculated and the then the distribution of forces in a diagrid structure will be explored.
Figure 10.7
Deflection of a cantilevered beam due to shear and bending

Figure 10.8
Lateral force resisted by the a truss

Figure 10.9
A moment force on a truss
Cantilevered beam

In high-rise buildings a governing parameter that should be satisfied besides the usual strength and stability is the stiffness that determines the deflection at the top. If the sway of a tower is too much people will get ‘sea’ sick and the tower may lose its strength because of metal fatigue. To calculate the deflection two cases need to be considered, shear and bending. The first due to lateral forces and the second because of the bending moment these forces generate.

This leads to the following formula \(^{[32]}\) (in which \(w = U\text{(total)}\)):

\[
w = \frac{ql^2}{2GA} + \frac{ql^4}{8EI}\]

Which can be simplified to:

\[
u(H) = \gamma H + \chi H^2\]

Diagonal Strength

A diagrid exists of only diagonals and does not need to have any vertical members; therefore all the forces are transferred via the diagonals. A big advantage is that because of truss action the forces will mainly be transferred as axial forces. This means that there will be hardly any moment in the beams and the stiffness is directly generated by the strain in the members.

The following assumptions are made to simplify the model and be able to validate the structure by hand:

- All forces are transferred as axial forces, and Hooks’ law can be applied;
- The floors behave as stiff planes, with no out of plane bending, spreading the loads equally and preventing torsion in the tower;
- The lateral forces are taken by the diagonals in “web” and the moment is carried by the diagonals in the “flange”;  
- When the Height to Width ratio is more than 5 only bending is governing

First the lateral strength will be checked according to a simple braced portal frame in which the columns are presumed to be infinitely large in bending stiffness (EI), no extra deformation due to column bending, and the horizontal beam is infinitely large in axial stiffness (EA), no extra deformation due to elongation or shortening of the top bar.
Figure 10.10
a possible outcome for a diagrid tower, 105m high, with crosses spanning 6 floors.

Figure 10.11
Moment and Force distribution for a cantilevered beam 105m high, 10kN/m

Figure 10.12
one slice of a square diagrid tower
When all variables are analysed and combined this formula can be derived[33]:

\[ u_h = \frac{F_n h}{2EA \sin \vartheta \cos^2 \vartheta} \]

This formula shows the relation between and the horizontal displacement and the lateral force and the angle of the diagonal.

Secondly if the structure is checked to resist a moment this is how it could be schematised (see fig. 10.9). When all variables are analysed and combined this formula can be derived:

\[ u_r = \frac{Mh}{BEA \sin^3 \vartheta} \]

These assumptions are then checked against models created in GSA and the results are very accurate.

**Diagrid Part**

In the next section the above derived formulae will be applied to get an idea about the axial forces in the members and to calculate the deflection. For this a tower is considered which is 105m in height, the base is 50,4m by 50,4m. The crosses span 6 floors, each 3,5m high and there are three crosses in both directions (see fig. 10.10, 10.11 and 10.12). In this case the moment is taken by the flange diagonals and the force is taken by the web diagonals. If the same formulae are applied as in the previous chapter estimations can be made for either the Area needed for the member or the deflection. In this case the lateral force is spread over 3 diagonals on either side of the building and the moment is spread over 3 diagonals as well. Now it is essential that the floor slab is stiff and strong enough to transfer the forces and thereby create equally distributed loads in all the diagonals.

To calculate the total displacement of this part the following formula can be used:

\[ u_{total} = \frac{VL_d}{2N_fEA_{web} \cos^2 \vartheta} + \frac{2ML_dH}{B^2N_fEA_{flange} \sin^2 \vartheta} \]

[33] See Appendix B for the derivation of the formula
And the stresses can be calculated with the following formulae:

\[
\sigma_{\text{web}} = \frac{F_h}{A_{\text{web}} 2 \cos \vartheta} \\
\sigma_{\text{flange}} = \frac{M}{A_{\text{flange}} B \sin \vartheta}
\]

These assumptions / formulae are then checked against the GSA model and the outcomes are nearly the same.

**Diagrid 5 stories**

Now all results from the separate parts are validated the next test is to see whether the estimations still holds when applied to a larger whole. Therefore in this section the diagrid existing of 30 floors with 5 crosses over the height is examined again. This time the deflection will be checked, and the stress in both the web and flange members will be checked. When these results are checked the ratio between shear and bending will become clear as well and will prove that once the height to width ratio is larger than a certain value only bending can be concerned to get an accurate estimation of the displacement at the top. The simplifications this time are:

- That there is no distributed load applied but a point-load at the top of every floor. Therefore the moment line can be considered straight;
- The floors are still ‘infinitely’ stiff for axial and bending loads.

(fig 11.13)

With help of the earlier derived formula we can now calculate the horizontal displacement at the top of the tower, but this time we have to calculate the displacement step for step at each floor since every angle due to bending and displacement due to lateral forces is different per floor. For displacement due to lateral forces these can be added and each time the load according to the shear should be applied (0(5th floor), 200, 400, 600, 800(1st floor))

\[
U_{h;\text{lateral};\text{total}} = \sum_0^5 U_{h;\text{lateral};\text{floor}}
\]
This leads to a displacement of 10.61 mm

The displacement due to bending needs to be calculated according to the tilting of the tower and then multiplied by the height.

\[ u_{k,b} = \frac{2ML_d H}{B^2 N_f E_{flange} \sin^2 \vartheta} \]

The total at the top is the sum of all \( u_{h;b} \) with the values at each floor (\( H = 105, M = 42000 \text{kNm} \) to \( H=0, M=0, N_f = 6 \)).

For instance at the first floor the rotation (1.5E-04) times the height (105m) is about 15.7 mm and then the displacement of the rotation of the other floors needs to be added as well, summing up to roughly 26.8 mm. This leads to a total of 37.4 mm, where GSA displays a deflection of 34.58mm. This means that the assumption of the corners staying 90° is not completely true, but accurately enough for now to check the results.

The stress in the web and flange members can be calculated via the before mentioned formulae:

\[ \sigma_{web} = \frac{F_h}{N_a A_{web} 2 \cos \vartheta} \]

Leading to a stress in the lowest members with \( F_h = 1000 \text{kN} \) and \( N_w = 6 \):
\( \sigma_{web} = 16.85 \text{ N/mm}^2 \) where this is again exactly the same value as with GSA in the middle diagonals.

\[ \sigma_{flange} = \frac{M}{N_f A_{flange} B \sin \vartheta} \]

Leading to a stress in the lowest members with \( M = 42000 \text{ kNm} \) and \( N_f = 2 \times 3 = 6 \):
\( \sigma_{web} = 22.45 \text{ N/mm}^2 \) where in GSA the value is slightly lower with 21.61N/mm².
Figure 10.13
The diagrid tower, 30 floors high with extra stiff floors, loaded at every floor.

Figure 10.14
The effect of a raised floor height, concerning daylight.

Figure 10.15
The relation of Volume vs Surface.
10.3.2. Conclusion

Although there are some differences in the displacement, stress or forces, calculated by the formula compared to the GSA models, these deviations can be considered small, in both cases, the scenario where the simplifications are made and the more realistic scenario. The above study contributed in getting acquainted with diagrid structures and knowing how the forces are distributed. This leads to the insight that the forces and therefore the stress increase heavily when getting closer to the bottom of the tower. An easy reduction of material could be made when the tower is divided in sections where the same member properties are applied instead of the same dimension everywhere.

This study also provided confidence in the way the GSA models are set up and therefore in the rest of the Thesis the approach will be to model the tower in GSA and work with the results derived from these models.

10.4 MEP

In the late nineteenth century, the early tall building developments were based on economic equations – increasing rentable area by stacking office spaces vertically and maximizing the rents of these offices by introducing as much natural light as possible. Today natural light is still a governing factor in design. The depth of a building is restricted by the amount of daylight that can penetrate. Most of the time the shape of the building, or the amount of glass in the facade, is decided by the orientation of the building. The radiation from daylight can heat up the building and this will lead to large mechanical ventilation installations and high energy consumption. The height of floor has a huge impact on the design of the building:

- The height will determine how much daylight will enter the building, and therefore the amount of rentable floor space;
- The height will determine the volume on the floors, which will determine the capacity needed for mechanical or natural ventilation;
- The height will determine the volume to surface ratio of the whole building, determining how much heat can be transposed;
- If the height of separate floors is larger, the whole tower will be higher, and therefore more façade (which is one of the most expensive parts of the building), leading to higher costs.

(fig 10.14)
(fig 10.15)
10.5 Conclusions

The named case study is used to get a feel of designing a structure and see all the values that are of importance for a diagrid design. Via this example it can be understood how a multi-disciplinary design project is influenced by several parameters. Again, some are local, some are shared or coupled. From the insights of a diagrid tower a list is distillate of factors that could be of importance to decide upon, for the success of the design. The list could be as follows:

- Height tower
- Floor height
- Kg steel
- m2 Floor space
- m3 Volume vs. m2 Façade
- % structure vs. facade
- Daylight
- Orientation (governing wind direction)
- Amount of nodes
- Amount of different elements
- Floors per Cross
- Maximum deflection (mm)
- Maximum axial force
- Needed capacity of mechanical/natural ventilation
- Angle of the diagonals (used to determine the optimal angle for strength and deflection)
- Etc.

These criteria can then be set to operators when the design has been run. By providing values to these operators, they will function as filters for the design, presenting all the outcomes that satisfy the criteria. It is very important to have the constraints set up in such a way that they can be mounted, adjusted and ‘played with’ to observe the changes in the outcomes produced. Although the constraints make up but a small part of the dashboard, they do determine the outcome and are vital in producing a final tool that is satisfying to use.
One of the big advantages of working with storing all data in a database is that all options are evaluated and therefore the minimum and maximum values are known. This makes it possible to scale all criteria of the alternatives and compare them with minimum loss by trade-off.

The previous chapters provide enough insight in how designing in a multi-disciplinary environment works and the case studies shows for a concrete example what to focus on. These results will be used in next part where the building of the tool itself is explained.
In the previous chapters all the ingredients are explored and together with the case study the reason why they are chosen is elaborated. In the next chapter the information that is gained will be used, but this time the focus will be on how the literature study and case study can be implemented in a tool. This tool should be designed in such a way that it can be of assistance for a structural engineer to make a well-founded ‘multi-disciplinary decision. The dashboard will be the tool via which design decisions and their interaction can be checked.
Figure 11.1
Django scheme, showing the process of the MVC-framework

Figure 11.2
Django sequence diagram
When setting up this program, it will be kept in mind to have everything set up in a generic way, so that it is reusable when the design changes into something totally different, such as a bridge, station, stadium, theatre, a tower with outriggers, etc. A simplified version of the earlier presented dashboards will be shown; all functions can still be implemented or extended.

The process described here is how a project can implemented, and at the same time can be set up. A workflow of the whole process and of a separate part will be given. At the end the results gained from the tool will be tested according to the rules derived with the case study via both; analysing the trends and a numerical check. Then an evaluation of the results and the use of the tool will be given.

### 11.1 Workflow of the process

The workflow of the process followed here is the same as the workflow mentioned in chapter 10. First a design was chosen, here a cylindrical diagrid tower. A first sketch and the case study, lead to parameters that are important to be variable. These are:

- The height of the tower
- The diameter of the tower
- The amount of crosses in the circumference
- The floors between the crosses
- The floor-height
- The profile of the diagonal
- The loads on the structure (deal-load and live-load)

From these parameters and the first idea of the tower a parametric model is set up.

Within the parametric (Grasshopper) model an export module to GSA (Salamander) is implanted. The next step is to start the Rhino, Grasshopper, Salamander model and open the Python editor. The Python editor is the file that runs the model, exports the data to excel and saves an image in a folder. Then all data should be converted to be stored in the database and be retrievable via the dashboard. This is done with Django as the MVC framework.

The scheme (see fig. 11.1) shows how the HTTP retrieves input from a user and provides input to the Django framework which then communicates with your own code and the data in the database before outputting the new values and
Figure 11.3
basic envelope of diagrid tower

Figure 11.4
adding floors to the model
implementing these onto the template, outputting on the browser in front of the user.
The workflow of my Django code can be presented more into detail via a sequence diagram, showing which steps are taken within program. The sequence diagram (see fig. 11.2) shows the interaction between the modules within pre-programmed Django, the views module (which exists of own code) and the templates (HTML).

11.2 Parametric model

Parametric models are very convenient when changes need to be made constantly. The power of parametric modelling can be exploited by changing the parameters while keeping the logic constant and storing the different created alternatives in the database. In this case a diagrid tower is set up with the help of Rhino and Grasshopper. In the model several phases can be recognised. In the next section screenshots will be provided of the grasshopper model. In the first image a simple envelope of the area will be sketched. In this case the variables are:

- Height of the tower
- Diameter of the tower

Once this is created (see fig. 11.3) the infill can be generated. Still via input all can be changed. In this case the extra input parameter is:

- the height of the floors (see fig 11.4)

The next step is to add the diagonals, to make the real diagrid. But even now it must still be the goal to be able to change the input parameters:

- The cross height (floors per cross)
- The amount of crosses in the circumference

These two values determine the angle of the cross and therefore as can be seen in the case study the strength and stiffness of the diagrid tower (fig 11.5).

The next step is to assign section profiles to the line model. Salamander will be used to achieve this. Here predefined sections can be assigned making the structure more realistic. The input parameter gets a new variable:

- The section profiles to the diagonals
- The wind load (N/mm2)
Figure 11.5
the changing of the cross height (since there are 12 crosses in circumference here, the cylinder turned into nearly a cylinder)

Figure 11.6
Assigning sections profiles to the wire-model.
• The Live Loads (N/mm²)
• The self-weight of the structure

see fig 11.6 and fig 11.7.

11.3 Calculation model

The next step is to export this file to GSA and calculate the solution this can be done via the Export to GSA component as seen in the Salamander model screenshot (see fig 11.8). Thereafter all results from Rhino, grasshopper and GSA can be stored in Excel, which in this case functions as the database. Now the parametric model is set up, all that has to be done is automate the process of changing the parameters and storing the data in the excel file. At the same time when the values are stored in the excel file a render should be made of the current view. This render can then be displayed on the dashboard providing extra visual information about the solution at hand.

In the new version of Rino 5 it is possible to run a python script and this script can control the grasshopper file, change the values and render an image. In this case there are 5 different sliders that can be modified:

• The first slider will be the slider changing the height, it needs a start, max value and an interval;
• The second slider will be the slider changing the amount of crosses on the circumference, it needs a start, max value and an interval;
• The third slider will be the slider changing the amount of floors the cross spans, it needs a start, max value and an interval;
• The fourth slider will be the slider changing the floor-height, it needs a start, max value and an interval;
• The fifth slider will be the slider changing the section profile, it needs a start, max value and an interval.

As pseudo (Pythonic style, with indentation) code that would read as follows:

```
RESET ALL SLIDERS AND VALUES
LOOP OVER THE SLIDERS
    FOR FIRST SLIDER IN SPECIFIED RANGE (START VALUE, MAX VALUE, INTERVAL)
        MODIFY SLIDER
    FOR SECOND SLIDER IN SPECIFIED RANGE:
        MODIFY SLIDER
    FOR THIRD SLIDER IN SPECIFIED RANGE:
        MODIFY SLIDER
    FOR FOURTH SLIDER IN SPECIFIED RANGE:
        MODIFY SLIDER
    FOR FIFTH SLIDER IN SPECIFIED RANGE:
        MODIFY SLIDER
RUN GSA
ADD 1 TO LINE TO PRINT EXCEL DATA
```
Figure 11.7
Using Salamander to assign sections, material, restraints and loading to the wire model.

Figure 11.8
the overall model
11.4 Database set up

First of all the database should be filled with data. As explained in the previous sections this data is generated via grasshopper. The whole parametric model and the results from every separate solution are stored in excel (see fig. 11.9 and fig. 11.10).

Excel is a flat database structure and not directly suited for the job at hand. But it is fairly easy to import the data via Python from excel into Django and store it in a (online) database. The class to convert the excel data and their columns to the columns in the database is located in the models file (showed later). This can be done with code like this:

```python
import the excel workbook from its location
loop over the rows
    if the row has a string value
        pass
    else
        check if these values already exists, if so retrieve and update,
        otherwise create
        read out column and store in class name
        continue with next columns until finished
        save class names in database
```

In the settings from Django the location where all this information will be stored is indicated. The same goes for the Model, View and Controller this is the big advantage of Django, most of the standard programming work and structure has already been laid out. All templates can be easily written in notepad with simple CSS and HTML code retrieving data from Django and extending the templates. In the case of this thesis a few important files are created: a setting file (A) and the models file (B).

A: The setting file
This contains what type of data base is used (in this case sqlite3) and where this will be stored. Furthermore all the locations to which will be redirected, such as

- The media root, redirecting to the images that will be displayed on the dashboard;
- The media url, assigning the right templates when there is redirected from a website;
- The template direction, finding the templates needed to display the right information.
Figure 11.9
Grasshopper to excel

Figure 11.10
Excel file with the data stored from the grasshopper model
B: The models file

Containing the different classes needed to be stored in the database in this case the ExcelData class, as can be seen in the database code:

```python
from django.db import models

class ExcelData(models.Model):
    filename = models.CharField(max_length=10)
    toverheight = models.DecimalField(max_digits=20, decimal_places=2)
    floorheight = models.DecimalField(max_digits=20, decimal_places=2)
    crossepforc = models.DecimalField(max_digits=20, decimal_places=2)
    crossexground = models.DecimalField(max_digits=20, decimal_places=2)
    angle = models.DecimalField(max_digits=20, decimal_places=2)
    floors = models.DecimalField(max_digits=20, decimal_places=2)
    crosso = models.DecimalField(max_digits=20, decimal_places=2)
    nodes = models.DecimalField(max_digits=20, decimal_places=2)
    sectionprof = models.CharField(max_length=20)
    kcom = models.DecimalField(max_digits=20, decimal_places=2)
    maxforce = models.DecimalField(max_digits=20, decimal_places=2)
    maxstress = models.DecimalField(max_digits=20, decimal_places=2)
    reactionforce = models.DecimalField(max_digits=20, decimal_places=2)
    deflection = models.DecimalField(max_digits=20, decimal_places=2)
    floorspace = models.DecimalField(max_digits=20, decimal_places=2)
    volume = models.DecimalField(max_digits=20, decimal_places=2)
    facadesurface = models.DecimalField(max_digits=20, decimal_places=2)
    lengthdiagonal = models.DecimalField(max_digits=20, decimal_places=2)
```

Every line functions as a column in the database, so here there are 19 columns storing the values from excel in different rows, all with their own identifier. The amount of rows is provided by the amount of options that are generated with the parametric model. In this database there are two different types, the CharField and the DecimalField.

The CharField is a character field, storing a string, in this case with a maximal length of 10 characters. The DecimalField is a double field storing numbers here the maximal length of the number can be 20 digits and the all numbers will be rounded to two numbers behind the comma if they are not integers.

11.5 Operators and filters

The operators are the interactive part of the dashboard. As can be seen from the previous sections the operators are generated by the Django code and correspondent with the values received from the grasshopper model and stored in the database. The operators are created in the views file, where all the coding happens. In case of the thesis the data mining is done here. First the data stored in the database is retrieved to generate the operators and assign there minimum, maximum and average value. This will create the start or interactive part of the dashboard, displaying the information on a template that can be broadcasted on the internet.
Figure 11.11
Operators as programmed in the tool

Figure 11.12
the final built dashboard, showing the operators, the graphs and the visual representation of the solutions derived by changing the parameters. Here there are 2016 options and the filters brought it down to 36 satisfying options.
A hierarchy could be very important if there is chosen to work with Ajax to make the process of choosing and editing filters and receiving the outcome via graphs or images (see fig. 11.11). When all the operators are provided with a value and the submit button is used to send the request to Django a new definition is started. Later on this can be automatically done with help of DAjax.

### 11.6 Dashboard

With the operators selected and edited the next step it to retrieve these values and draw graphs from it. These graphs have to be predefined and contain the information requested.

The graphs on the dashboard (see fig. 11.12) can then be edited via changing the data in the operators over and over again, whilst analysing the graphs and narrowing down the solutions space while keeping the possibility to change everything if extra insights are gained or information from other disciplines influence the choices.

The interaction between all the technical coding and the visual dashboard is done via HTML pages, but these are driven by theUrls file that communicates with the browser whenever tasks are given by Django or retrieved from the template the URLS file directs them to the right place and right definition. With the help of Django it is possible to have a system that runs a database, formulates data queries via simple pythonic code and output webpages edited by simple HTML and CSS code.
The above showed tactics, technology and information is all used to make the dashboard and come up with a decision supporting tool for Multi-disciplinary optimisation problems. The next phase will be testing the program and validating the results the program outputs. After that an evaluation of the program will be given.

11.7 Testing and results

The tool will be tested here to see whether it does what it should do, provides useful information and is easy to use. Before the actual tool can be tested first all the results need to be created. The grasshopper model shown in the previous section is selected and the parameters that can change are chosen and set:

- **Towerheight:**
  - start value 30 meters
  - Maximum value 90 meters
  - Interval 20 meters

- **Crosses on circumference:**
  - start value 3
  - Maximum value 9
  - Interval 1

- **Floors spanned by cross:**
  - start value 1
  - Maximum value 6
  - Interval 1

- **Floorheight:**
  - start value 3,5 meters
  - Maximum value 4,1 meters
  - Interval 0,2 meters

- **Section profile:**
  - HE120.A
  - HE280.A
  - HE550.A

The other values that are set but not variable are:

- Diameter = 13 meter
- Windload = 2 N/mm²
- Selfweight of the structure
When these selections are applied and all the parameters are changed and calculated a set of 2016 options are generated. This was done on a fairly old laptop within roughly 22 hours. But the larger the model the longer it takes. The last options (90 m tall tower) took about 2 minutes to generate and calculate, whereas the first options (30 m tall tower) could do 6 options in a minute. All the data was then written to an excel file and the images where stored on a hard-disk. 

After all the results are generated, Eclipse (the coding program) can be started and phase two (as seen in figure 9.1) of the program can start.

In this example only the structural elements are considered, from the case study it shows that especially the angle is of importance for the strength of the diagrid. Therefore there is chosen to have three graphs via which you can see the influence of the angle against the maximum stress, the tonnage steel and the deflection. The other graph displays the maximum stress against the steel tonnage.

With the next images one of the ways of testing the results and checking if the data is corrupted in any way is showed.

The first screen that will open shows all the data (see fig 11.13). This shows all the 2016 options, resulting in very crowded graphs, although the first trends are visible, the only thing that can be concluded is that there are no values totally off, everything is roughly in the same range.

In the graph all the dots have a number assigned to them; this is the ID of the graph, explaining of which parameters the option exists.

The ID is as follows: 90_9_4_39_9

- The first number is the height of the tower (options are 30m, 50m, 70m and 90m)
- The second number is the crosses in circumference (options are 1, 2, 3, 4, 5, 6, 7, 8 and 9);
- The third number is the floors per cross (options are 1, 2, 3, 4, 5 and 6);
- The fourth number is the floor height (options are 35, 37, 39 and 41 which translates to 3.5m, 3.7m, 3.9m and 4.1m);
- The last number is the profile number according to the slider in grasshopper (options are 1, 9 and 17 which translates to 1 = HE120.A, 9 = HE280.A and 17 = HE550.A).

Now the first filter can be applied, in this case there is chosen to select the maximum stress and set the maximal allowable value to 235 N/mm2. This leads to 395 satisfying results.
Figure 11.13
the final built dashboard, showing all results after initialising the dashboard.

Figure 11.14
the final built dashboard, showing the results after filtering to a maximum stress of 235N/mm²
For the screenshot, the maximum stress operator has been moved up to be checked earlier and to be displayed on the screen (see fig 11.14).

Although there are substantially less options now, it is still difficult to see any trends or understand what is going on.
A next step can be to decide the tower has to be at least 70 meters high. This leads to the next images with 78 satisfying results (see fig. 11.15).

With these graphs it can clearly be seen that two options need far more steel to satisfy the results. These options are towers 70 meters high, 9 crosses in the circumference, a cross only spanning 1 floor, floor-height of either 3,7 or 3,9 meter, and profile of HE550.A.
All the other options have crosses spanning more floors, and therefore less crosses, although longer diagonals there will be less steel in total. Furthermore there can be seen that there are 16 possibilities for a tower of 90m high, all with 8 or 9 crosses in circumference, and 62 options for the 70m high tower with either 6,7,8,or 9 crosses in circumference.

So far the program behaves well and more selections can be made, by setting most of the operators to governing values only a few options stay.

Now there has been some shuffling of the operators and the values are set (fig 11.16). The tower has to be 70m high with a floor height of 4.1m, crosses spanning more than 5 floors, a minimum of 6 crosses in the circumference, and the angle has to be more than 40 degrees, if the deflection is restricted at 170mm this leads to 4 options that are still valid

So far searching through the data has not displayed anything that does not work, nor present values that cause suspicion about the parametric model, analysis models or the web framework. The only flaw in the system seems to be that only integers can be entered and point nor comma separated values are taken as input.
The next step is to validate the results and see if the values received could be considered correct.

11.8. Validating the results

This section is to validate the results retrieved via the dashboard. In this case a cylindrical diagrid tower is considered. The case study derived a formula to calculate the deflection based on wind load.

\[
\begin{align*}
    u_{\text{total}} &= \frac{VL_d}{2N_w EA_{wet} \cos^2 \theta} + \frac{2ML_d H}{B^2 N_f EA_{flange} \sin^2 \theta}
\end{align*}
\]
Figure 11.15
The final built dashboard, showing the results after filtering to a maximum stress of 235N/mm², and higher than 70 meters.

Figure 11.16
The final built dashboard, showing the 4 resulting results, after a many criteria.
The case study was performed on a squared tower, but the principles found there should be in line with a cylindrical tower. From the formula it can be seen that to deflection depends on:

- The load, causing shear and moment forces, in the model here these are kept constant, although change in height of the tower influences the forces, all equally high towers have the same loads applied (V & M).
- The length of the diagonal, being a function of the crossheight, the angle and the amount of crosses in circumference (Ld).
- The area of the diagonal (Aweb & Aflange)
- The amount of crosses (Nw Nf)

With this in mind first step is to only compare towers with equal height, so set the operator that it will only display the 90m high tower, with a set floor height, 4.1m. This leaves 126 valid options (see fig 11.17).

The first trend that can be spotted now is that the stress reduces every time the amount of crosses is in the circumference increases; this is the N factor in the formula (fig 11.18). The same trend can be spotted in the bottom left (green) graph displaying the angle and deflection. If this is explored some more and the amount of floors per cross is filtered as well, the dashboard provides 21 solutions (fig 11.19). Now all four graphs will be considered and will be explored to see if all governing elements from the formula can be validated (fig 11.20).

From the above graph two things can be noted, the first is that again the amount of diagonals (N) do matter for the maximum stress, but they also influence the angle, if there are more diagonals on the circumference the angles become steeper. (fig 11.21).

A thing that can easily be noted from the graph is that if the profile changes from a HE120.A to HE280.A and HE550.A the maximum stress and deflection reduces, but the tonnage steel rises.

(fig. 11.22)

To get more insight into how the angle is influenced, a plot is made of a 90m high tower with 9 crosses in circumference and 3.9m floor height. This way the angle only changes if the cross spans more floors. Now insight can be gained in the change of angle and the stress or deflection caused by this change. This results in a new graph set.

(fig 11.23)
Figure 11.17
the final built dashboard, validating the 90m high tower, with 4.1m high floors.

Figure 11.18
Spotting the first trend based on the amount of crosses (N).
Here suddenly something strange pops up. There should be 6 peaks in the graphs (1,2,3,4,5,6) but somehow it only shows 4 peaks, the other peaks are at the same angle, therefore there should be something wrong with the program. Another strange thing is that in excel the angle of the cross does not behave the way it should.

(fig 11.24)

Here can be seen that in the highlighted column the angle does get steeper if the floor height increases from 3500mm to 3700mm, but it becomes flatter if the floor height changes to 3900 and then steeper to 4100mm but still not as steep as at 3700m. This does not make sense because if the floor gets higher and the rest stay the same the angle should get steeper.

After looking into the grasshopper model, it appears that the angle is calculated with the diagonals on the top. Since the height of the building in floors might not always be dividable by the amount of floors a cross spans, this causes two problems. One this leaves a left over in the amount of floors a cross can span (for instance 25 floors and crosses that span 4 floors leaves a leftover of 1 floor (25/4 = 1.25). Secondly since the tower will always be 90m high, the last floor can become higher since the left over in height will be added to the top floor (90/4.1 = 21.95, so the last floor will be 8m high instead of 4.1m.

This does explain why the values are close to the normal values:

• 90m/3,5 = 25.71 floors = 24 floors of 3.5m and 1 of 6 meters
  o In case of 1 floor per cross the angle should be \(\tan(\alpha \times 6/3.5)\) times to large;
  o In case of 2 floors per cross the angle should be \(\tan(\alpha \times 6/7)\) times to small;
  o In case of 3 floors per cross the angle should be \(\tan(\alpha \times 6/10.5)\) times to small;
  o In case of 4 floors per cross the angle should be \(\tan(\alpha \times 6/14)\) times to small;
  o In case of 5 floors per cross the angle should be \(\tan(\alpha \times 20/17.5)\) times to large;
  o In case of 6 floors per cross the angle should be \(\tan(\alpha \times 6/21)\) times to small.
Figure 11.19 elaborating the first trend based on the amount of crosses (N).

Figure 11.20 elaborating the first trend based on the amount of crosses (N).

Figure 11.22 elaborating the first trend based on the amount of crosses (N).
Figure 11.21 showing the amount of crosses (N), (3, 4, 5, 6, 7, 8, 9).

Figure 11.23 elaborating more to discover the influence of the angle.
Figure 11.24 elaborating more to discover the influence of the angle, in the 'database'.

Figure 11.25 The dataset when the filters are set to 90m high, 9 crosses round and 3.7m high storeys.
This would lead to the following angles:

- 21.5 instead of 34 degrees
- 38.2 instead of 34 degrees
- 49.7 instead of 34 degrees
- 57.6 instead of 34 degrees
- 63.1 instead of 66 degrees
- 67.0 instead of 34 degrees

Since the values now are not correct the real validation can’t be done, but the trends except for the angle can still be of valuable outcome. This leaves validating the influence of the area on the stress, deflection and tonnage steel. With this error in mind the program is used to find a value that does show 6 peaks, or the most possible.

After a search with changing both the height of the tower and the floor height it gives the following results:

- **Tower 90m height, with 9 crosses in circumference:**
  - 3,5m gives 2 peaks
  - 3,7m gives 5 peaks
  - 3,9m gives 4 peaks
  - 4,1m gives 2 peaks

- **Tower 70m height, with 9 crosses in circumference:**
  - 3,5m gives 4 peaks
  - 3,7m gives 4 peaks
  - 3,9m gives 3 peaks
  - 4,1m gives 3 peaks

- **Tower 50m height, with 9 crosses in circumference:**
  - 3,5m gives 3 peaks
  - 3,7m gives 2 peaks
  - 3,9m gives 5 peaks
  - 4,1m gives 5 peaks

- **Tower 30m height, with 9 crosses in circumference:**
  - 3,5m gives 3 peaks
  - 3,7m gives 3 peaks
  - 3,9m gives 3 peaks
  - 4,1m gives 3 peaks

Here there is chosen to continue the validation with the Tower 90m high and a floor height of 3,7m.

In Figure 11.25 the red circle shows where the double numbers are.
The angles of the diagonals are and should be:

- In case of 1 floor per cross the angle is 28.9 degrees and should be 22.6
- In case of 2 floors per cross the angle is 44.0 degrees and should be 39.8
- In case of 3 floors per cross the angle is 54.1 degrees and should be 51.3
- In case of 4 floors per cross the angle is 60.9 degrees and should be 59.0
- In case of 5 floors per cross the angle is 60.9 degrees and should be 64.3
- In case of 6 floors per cross the angle is 69.2 degrees and should be 68.2

The angles with 4 or 5 floors per cross give the same angle but should be different, this is the same as can be found on the dashboard.

A trend that can easily be spotted and is still relevant, is the change in deflection or stress caused by enlarging the area of the profile. In this case there are 3 profiles that can be selected:

- HE120.A with an Area of 2533mm²
- HE280.A with an Area of 9726mm²
- HE550.A with an Area of 21170mm²

- From profile 1 to 2 is a multiplication of (9726/2533 = 3.84)
- From profile 2 to 3 is a multiplication of (21170/9726 = 2.18)
- From profile 1 to 3 is a multiplication of (21170/2533 = 8.36)

If the formula is looked at again:

$$u_{total} = \frac{VL_d}{2 N_w E A_{web} \cos^2 \theta} + \frac{2 M L_d H}{B^2 N_f E A_{flange} \sin^2 \theta}$$

According to the program the graph should be as shown in fig 11.26. It should mean that the deflection difference should be in the same ratio. If the series in the blue circle are tested the values are:

- 90_9_6_37_1; the deflection should be 170080mm (20.7 times larger then profile HE280.A)
- 90_9_6_37_9; the deflection should be 8229mm (6.4 times larger then profile HE550.A)
- 90_9_6_37_17; the deflection should be 1289mm (still much too large if the maximum allowable deflection would be L/500 = 180mm)
The red circle which has 5 floors per cross is the value that does not have a peak at the right spot, but the values are much closer to each other.

- 90_9_5_37_1; the deflection should be 2169mm (3.84 times larger than profile HE280.A)
- 90_9_5_37_9; the deflection should be 565mm (2.18 times larger than profile HE550.A)
- 90_9_5_37_17; the deflection should be 259mm (still a little too large if the maximum allowable deflection would be L/500 = 180mm)

Here the ratio between the deflections is exactly as would be expected from the formula. If from this section all comparisons are made the results are:

- 90_9_1_37 (4.00 and 2.35)
- 90_9_2_37 (6.14 and 2.43)
- 90_9_3_37 (12.5 and 3.33)
- 90_9_4_37 (17.32 and 4.67)
- 90_9_5_37 (3.84 and 2.18)
- 90_9_6_37 (20.67 and 6.39)

Here a problem is caused by the export from grasshopper to GSA. Therefore I looked at the GSA files there where the difference between the deflections did not correspond with the ratio between the areas of the section properties. Here it appears that some elements did not copy across (see figure 11.27).

The trouble is probably caused by deleting double nodes via a component called the design doctor, this component deletes coincident nodes. Somehow it does not always merge the beams with the nodes and sometimes beams get deleted, causing a corrupt model.

The setup of the parametric model can be assumed to be correct, but the Salamander component is corrupted. Every time when the size of deflection is not in line with the ratio between the sections, the GSA model does miss some beams.

The values of the non-corrupted files are much more the same as should be expected from the hand calculations. Unfortunately this are one some options. See fig 11.27.

The developer of Salamander (Paul Jeffries) has been contacted and he will try to figure out why this happens and will try to resolve the problem. Again this makes validating the design case impossible, but the tool proved to be a perfect way for spotting malicious files.
Figure 11.26
The dataset when the filters are set to 90m high, 9 crosses round and 3.7m high storeys. Here two sets are highlighted (90_9_5_37 and 90_9_6_37).

Figure 11.27
Some elements are not copied across, causing a much larger deflection.

Figure 11.28
The dataset when the filters are set to 90m high, 9 crosses round and 3.7m high storeys. Here the tonnage steel is set out to the angle of the diagonals.
Unfortunately yet another thing can be noted, with a steeper angle the tonnage steel reduces, this will cause higher stress in the resulting material because the structure will be opener. But the amount of steel and how it changes is very strange. This should only be a little difference but here the difference is very large.

It is known that profile 9 (HE280.A) uses about 3.84 times more material and profile 17 (HE550.A) uses 8.36 times more material than profile 1 (HE120.A), this roughly should be the difference in steel tonnage as well. But if we look at the graph the tonnage is about the same between these options. The difference in weight caused by the change in crosses spanning different floors is immense (it could be in the order of a few percentage, but should not be double as can be seen here, fig 11.28).

Again looking back in the Grasshopper model, it seems that there is trouble with the salamander component that should calculate the amount of steel. To model the diagrid as realistic as possible the floors at the ends of the crosses are modelled as a spoke wheel with a strong rim. Although it should not add this weight to the calculated material it does mysteriously add the material. This means that crosses only spanning 1 floor have a lot of material in the horizontal plane, which should not be added. See figure 11.29.

The grasshopper file did have some errors, some in geometry (which can be solved, since all the errors are understood) and some in the implementation via Salamander to GSA (this need to be communicated with Paul Jeffries). Not all options that you need in GSA are available in this build of Salamander and the many workarounds now prove to be error prone. This leads to a different outcome from the hand calculations and the ‘automated’ model.

Another trend that was expected to show is the effectiveness of the angle and an optimum in this to be somewhere between 40 and 60 degrees steep diagonals. Now the graph looks like figure 11.30, with the expected line drawn in it as well. With the generated graph it can be seen that some of the geometry and settings in the model are not correct. The exporting to GSA with the interoperability between Salamander and GSA is corrupted and causes the wrong data and therefore the wrong trends. See fig 11.30.

The tool proves to be very valuable for searching trends and analysing a parametric model.

The complex layout of the Grasshopper model does not have to be fully understood by a third party to use this tool to effectively spot errors in the model. Before fine tuning, the model should be correct and this tool helps checking the validity of the model.
Figure 11.29
the difference in material between the two options.

Figure 11.30
Angle plotted against the stress and an extra line drawn by hand for the expected trend (orange dashed line).
11.9 Evaluating the results

The evaluating of the tool went slightly different than expected. The tool proved to be very successful and notified errors in the analysed model that I was not aware of. The behaviour of the program is very agile and capable of searching through a large dataset of (in this case 2016) options. The operators work very easy, everything can be changed and multiple ‘active’ operators can be added or removed and the whole will be analysed again within a few seconds.

Via the graphs and the gained knowledge from the case study, scenarios could be zoomed into and the visual representation of the data proved to be a very strong tool in analysing the results. The visual database on the lower part of the dashboard works very well to get an idea of the layout of the tower and see the difference between the options. By clicking on the graphs or an image it will appear on a new webpage and it can be studied into more detail. Thereby it is very easy to change the graphs to different values and analyse different relations.

Although the actual structural results are not as expected (see the section about validating the results and the error in the Salamander component, fig. 11.31), the tool was very useful to trace down the errors and gain insight in what to change in the parametric and analysis model.

Having a program that does not do exactly what you expect, might be better to test the tool with, you have to dig really deep and find the errors in the model. When you have a model that behaves exactly how you think it should, you might pay less attention to it and assume the rest is right as well. This can be seen from the first few assumptions in the testing section. Here the first trends seemed to be correct and it would have been easy to neglect the other relations and really explore how the angle is influenced and what the results are.

Again this process proves that although the computer can calculate a lot of alternatives, there still should be a specialist behind the dashboard to fully comprehend the results and validate these.

The results of this basic tool are very promising, but there is still a lot that needs improvement. First of all Salamander needs a lot of extra developing to really do what it supposed to do.
Figure 11.31
Dashboard showing graphs with different values, floors, floor height, crosses, kg-steel, stress, normal force, deflection and angle.
Secondly the tool exists of many non-coupled components. Therefore the process is a bit scattered. Designing and then building the parametric model will probably always be separate steps. The analysing, generating results, storing the results in the database and implementing the different variables as new columns in the database and present these on the dashboard is done manually in this example. It is possible to code all this together with the help of python and control all the steps after building the parametric model with the analysis cases on the dashboard. But for now the system is still graceless and many different steps have to be taken to get to results:

- Open Rhino, Salamander, Python Editor and Excel
- Run the Python editor (for the 2016 examples this took 22 hours, which is a long time)
- The data will be stored in the Excel file, which has to be saved as a 97-2003 version
- The images are stored localy on the computer and need to be relocated at the location stated in Eclipse (Django settings)
- Eclipse has to run (this is where Django is coded with python)
- Eclipse has to read in the Excel database and store this in a Sqlite3 format. This is done statically, by manually creating the column names in a class. Then the database has to be created and synchronised before it can run
- Than the tool runs on a localserver, not published on the web yet.

These steps are laborious to do but provide good insight in the whole ‘design/develop’ cycle of the tool. It would be much better though if all programs could be run from one centrally stored location, in this case Eclipse, and code would execute all the above mentioned steps so you only have to access Eclipse and run the manage.py file. This would enhance the user-friendliness.
CONCLUSIONS, RECOMMENDATIONS AND EVALUATIONS
12.1 Conclusion

Design considerations
The building sector is a sector in which roles and tasks are uncertain, with the ill-structured nature of a design problem then getting aggravated by this. Multi-parameter and multi-discipline solutions are often hard to reach in an ambience of mutual understanding. Nowadays the structure of the process in building design can have a high impact on the relations between the different parties, forcing people away from their own specialisations into taking on a more multi-disciplinary designer role. Being able to convince the other parties involved and to arouse emotions is very important, and all disciplines should have similar or the same tools to achieve this. Most of the time, the architect is the most outspoken person who is backed-up by beautiful visual representations of the design. This causes a lot of trouble in realising the project, the dreams that people chase become unrealistic for the amount of money available and slowly the design will be downgraded. Although sometimes ambition surpasses budget, working together from the start and understanding the restraints for the other disciplines can help in achieving dreams.

Performance
The decision making tool presented in this thesis shows that there are opportunities to visualise all the data that is generated by the team on a dashboard. The dashboard proved to be a very effective tool in making well-considered choices. The parametric system works very well for generating many alternatives, but special care should be taken that the model is correct and that all components work as they should. The tool has to be used by someone who is skilled in this discipline and will only help spotting trends or flaws in the results. The tool will not make the decision, but merely support the user. The current tool is very user friendly, easy to use or change and with the results being rapidly displayed on the dashboard. This makes it possible to search through a large dataset and compare the different values obtained. In spotting trends the graphs function very well, and the easiness with which they can be changed and adjusted enables the user to freely switch back and forth between all the possible parameters. The graphs can be deceptive however; a non-skilled user might easily get sidetracked and find all data to be overwhelming. But once the basics behind the structure are understood, playing with the parameters allows one to reach an optimal solution with a greater level of efficiency and success.
Advantages
The advantage of generating every possibilities available is that we enlarge our knowledge about a specific design, which can then be coupled with the idea of “what we do is based on what we know” and enables us to make thoroughly weighted choices.
Generating every alternative available is also very useful in exploring options that might not have come to mind if only ‘sensible’ options were to be considered based on hand calculations.
Another big advantage of all the options available being generated is that since the minimum and maximum value are brought to light, you can easily scale all the parameters and thereby prevent trade-offs or scale problems in searching for the ‘optimum’.

Outlook
The current dashboard can be changed to set up in and set up to work in a generic way with widgets and templates, thus adapting the layout to the personal tastes of the user. In this way different disciplines can use the widgets that suit them most and explain their case to the rest of the team.
The widgets that have been designed in this thesis are only a beginning, the opportunities for development are endless and can range from graphs, to plans, videos, textual data, renders or time schedules. At present such developments have not been carried out, however the experience gained in the thesis work provides strong evidence that it will be possible in the future.
The weighting of the parameters can be done on the spot or afterwards, making the decision process more transparent and enabling real-time changes to be made with a whole team of different disciplines present.
Interoperability between the analysing and geometrical programs is essential for future use of this system, but and this aspect is very important for the future development of the tool. It will provide multi-disciplinary teams with a lot of crossover information and enable them to browse through it in a visual and understandable way. It will be a powerful tool to use when explaining concepts, ideas and developments to other people, without those people necessarily needing to know all the complexities involved in the process and background work.

This way of analysing large datasets will make the design process more tangible, and once this is understood better it will be easier to convince people that the outcome is the highest possible value for money. But it also leaves the possibility that the final say is not put down to budget, with the options being compared based more on their individual merits instead of purely on their costs. This culminates in beautiful projects that incorporate and synthesise the various specialities involved, both in process as well as in the final result.
12.2 Recommendations

There are many things that still can be explored; this is only the start of a new way of thinking. But a few recommendations are given to enable further research to be carried out on the topic.

- First the framework has been built in a very basic manner, but the implementation of real multi-disciplinary optimisation has not been implemented. This is mainly because of the lack of discipline specific knowledge about the software programs or parameters of influence. To make these accessible for the system it is important to have or develop interoperable software that links the geometry to analysis and store the data in a database.

- Develop the system to have all the programs combined into one single program and store everything directly in the database, coupled with having the program controlled via the dashboard. Another thing that it will enable is that it will then be possible to run the program virtually in real time, instead of several processes that each take time to complete. By running on a cloud basis, the need of having all the memory intensive and expensive software on the device that shows the dashboard is not necessary anymore. This would enable the programme to be used on mobile devices such as a mobile phone or iPad.

- Different disciplines need different dashboards. The same framework can be applied, but the display can hold different criteria. In getting around this problem different widgets have been designed, but these are currently only theoretical and still need to be made into reality.

- Implement different ‘optimising’ algorithms such as:
  - The Monte Carlo analysis to find the sensitivity of specific operators, finding a fitness score for the alternatives.
  - Implement a Pareto front to find all the solutions that score best compared to the other options. This way money would not be the main drive behind the design, but could be set as a parameter within the design.

- Find a way to objectively capture architectural or visual preferences. Algorithms exists that can calculate ‘beauty’ for example according to symmetry aspects.

- Make the program intelligent by using collective intelligence, analysing user preferences to increase the efficiency in searching for ‘optimal’ solutions.
12.3 Evaluation

First it is very important to acknowledge that technology will enable us to develop these kinds of software. This is only a start however, and the interoperable software does not yet fully exist. It is ever closer to becoming a reality, with an example being the plug-ins that can currently be found for Grasshopper, which all enable various types of process to be carried out that are focused on different disciplines or users. A further very important point to note is that it is a decision supporting tool and by no means does away with the need for people to design a project, transform it to a parametric model and then with this tool verify the results with their own knowledge and expectations.

The thesis has brought to light several very interesting topics, with a number of them being still only at the very beginning of a whole process. To many sub areas attention could and should be spent. At the moment the tool misses some real interaction between the different disciplines. Although the concept framework has been developed and extra information can easily be added to the database and added to the dashboard, this thesis has only briefly addressed the multidisciplinary aspects such building physics, lighting design, MEP, sustainability, etc. The concept of the tool however has been elaborated, where the data to be fed to the tool can be arbitrary and does not change anything of the concept. The widgets prove to be a very important aspect to allow the dashboard to be personalised to the needs of the user. The other advantage of the widgets is that it is possible to make adjustments that dig deeper into the knowledge gained, which in turn build on previously gained insights and new ways of searching through reduced datasets.

To build a mock up it was decided to make the tool primarily for the structural engineer, with the input from other disciplines. This enables the structural engineer to slightly change their perspective and become more of a multidisciplinary designer. As a pure structural optimisation tool, this would not be very interesting since experience (‘tacit’ knowledge) will always be faster and the preferred choice. But the tool would be highly beneficial for combining knowledge from different fields. In this case the build-ability and aesthetics are considered. The option to scroll through design solutions provides a visual tool that can serve two purposes, first to have an idea of the sexiness of the design and secondly by looking at the solutions, trends, crossovers and trade-offs can be spotted and a feel will be obtained about the design and the desired outcome. Additionally from looking at the graphs other preferences will come to light (for example the amount of nodes or the of ratio steel to stress) and the way the team makes a decision can be addressed. Everyone makes a different decision at a different time and in a different group of people. Therefore I’m not worried
that the tool will lead to similar projects, human choices will still prevent unity
generated by automation.

On a personal level, the whole process of the thesis has been very inspiring and
educative. A lot has been learned about design processes and how different types
of knowledge can be applied. But also the difficulty within the building process
such as the lack of repetition, the coarse grained methods providing solutions in
an early stage at the design. The importance of rules of thumb, coupled with not
just applying these rules but knowing why and when they should be used, and
how other parties are affected by the decisions made. The whole design process
is a process that inspires a lot, how collisions can be prevented, whether they
are because of a construction point of view, strength, beauty, sustainability etc.
In the end the solution should be the best value for money that is possible and
even exceed the objectives that were the points of departure.

Besides the design process a lot has been learned about the advantages that can
be achieved via the smart use of the computer and computational technologies.
At the start of the thesis there was hardly any programming knowledge, but at
the end some of the basics were understood. This enabled thesis concept to be
considered on a more practical basis, whilst also on a more conceptual note.

Besides the computational aspect the experience of programming has ingrained
in me a structured manner of tackling a problem, which could easily be applied
to any other task faced. A good example is the analysing of the diagrid tower
in which a small part is extracted and finally is replaced in the whole solution.
Applying a small test, and once these work embedding them in the total solution
could be seen as a typical programming solution. The fun of analysing the
diagrid, puzzling with the way loads flow and what effects the structure most
is something that was highly enjoyed. Once the structure is better understood
the loop can start all over again, new parameters and errors needing amending
are found. To see design as an ill structured problem could be considered true,
but the process of solving it can be highly structured. A massive drawback is off
course the malfunction of the Salamander components, leading to disappointing
results when validating the tool.

The last aspect that has been addressed is the importance of data. We live in an
age where data doubles roughly every 18 months compared to the two and half
centuries it took in the renaissance to double. The use of and accessibility to data
has exponentially grown and this should definitely be used to our advantage.
Even though all projects are unique, experience and preferences make it possible
to find the right data and push the design a step further. The tool described in
this thesis can therefore be seen as the first step into a new direction of cutting
edge technology in comparing data and combining this with experience, thus
providing solutions in a manner that exceeds the capabilities of the human brain
alone by combining all the possibilities available.


Coenders, J.L.. 2007. Interfacing between parametric associative and structural software, Delft University of Technology, Arup Amsterdam.


Holzer, D.C.C. 2009: Sense-making across collaborating disciplines in the early stages of architectural design, Melbourne, RMIT.


Moon,K. 2008. Material-Saving Design Strategies for Tall Building Structures, School of Architecture, University of Illinois at Urbana-Champaign


APPENDIX
APENDIX A

CANTILEVER-BEAM, BENDING AND SHEAR
To calculate the value of the deflection at the top of a cantilever beam, first some formulae for either case (shear and bending) should be derived [36].

Shear due to lateral forces can be understood by the following figure;

![Figure A1](image1.png)

From the above figure it can be seen that to get equilibrium you need:

\[-D + q\delta x + D + dD = 0\]

\[q\delta x = -dD\]

\[q = -dD / \delta x\]

Because of bending the next situation can be considered:

![Figure A2](image2.png)

\[-M - D\delta x + m\delta x + M + dM = 0\]

\[m\delta x = D\delta x - dM\]

\[m = D - dM / \delta x\]
These shear and bending forces cause the element to undergo a rotation along the x-axis of $\alpha$ and the section rotates over an angle of $\phi$. The theory here is based on the hypotheses of Bernoulli that states that straight sections stay straight and the occurring rotations stay perpendicular to the axis of the beam. This means that the strain in the material is linear to the height of the beam.

This is only true if the value of $\frac{\delta w}{\delta x}$ is small so that $\tan(\frac{\delta w}{\delta x}) = \frac{\delta w}{\delta x}$.

In linear elastic materials there will be a linear relation between the shear and deformation, the proportionality constant which can be called the shear stiffness with the symbol $GA$.

For the shear stress in a section Hooke’s Law will hold, here $G$ is the shear modulus of the material. When the shear stress is constant over the area the next formulae can be derived:

$$D = A\tau$$
$$\tau = G\gamma$$
$$D = GA\gamma$$
$$D = GA(\delta w / \delta x + \phi)$$
The forces in this element also make the fibres elongate, the more you get to the lower part the more they get stretched (increasing in z-direction). This can be shown in the next image.

![Figure A4: Stretching of the fibres](image)

\[ \frac{\delta \varphi}{\delta x} = \beta \]

\[ \varepsilon = z \frac{\delta \varphi}{\delta x} = z(\frac{\delta \varphi}{\delta x}) = z\beta \]

To calculate the moment in this element together with Hooke’s law stating that \( \sigma = E\varepsilon \), the next steps can be taken:

\[ M = \iint_A z \sigma \delta A = \iint_A E\varepsilon z \delta A = \beta E \iint_A z^2 \delta A \]

If the section is homogenous in material and shape this leads to:

\[ M = \beta EI \]

\[ M = EI(\frac{\delta \varphi}{\delta x}) \]

From all the formulae derived above these can be combined into the next four that are important for a cantilevered beam subject to bending and shear:

\[ q = -\frac{\delta D}{\delta x} \]

\[ m = D - \frac{\delta M}{\delta x} \]

\[ D = GA(\frac{\delta w}{\delta x} + \varphi) \]

\[ M = EI \frac{\delta \varphi}{\delta x} \]
When looking at the cantilevered beam the next preconditions can be found:

In this case there is no additional moment in the beam so m=0

The formulae can then be rewritten to:

\[ D = -qx + C1 \]
\[ M = -\frac{1}{2} qx^2 + Clx + C2 \]
\[ \varphi = (-\frac{1}{6} qx^3 + \frac{1}{2} Clx^2 + C2x + C3) / EI \]
\[ w = \frac{-\frac{1}{2} qx^2 + Clx - \frac{1}{24} gx^4 + \frac{1}{6} Clx^3 + \frac{1}{2} C2x^2 + C3x}{GA} + \frac{1}{EI} + C4 \]

Filling out the preconditions for L(x=0) and L(x=l)
This leads to:

C3 = 0
C4 = 0
C1 = ql
C2 = -\frac{1}{2} ql^2

Leading to a deflection at L(x=l):
\[ w = \frac{ql^2}{2GA} + \frac{ql^4}{8EI} \]
Braced Frame

B.1. Horizontal Displacement

Here the lateral displacement due to a horizontal force is calculated. This can be worked out to:

\[ F_h = 2F_d \cos \theta \]
\[ F_d = EA \varepsilon \]
\[ \varepsilon = \frac{e_d}{L_d} \]
\[ e_d = u_h \cos \theta \]
\[ L_d = \frac{h}{\sin \theta} \]

This can be worked out to:

\[ u_h = \frac{F_h h}{2EA \sin \theta \cos^2 \theta} \]

In this test case:

- \( E = 205000 \text{ N/mm}^2 \)
- \( h=b=5 \text{ m}, \text{ therefore } \theta=45^\circ \)
- the diagonals are chosen to be CHS 102*5.0
  - \( A = 1520 \text{ mm}^2 \)
  - \( l = 1770000 \text{ mm}^4 \)
- \( F_h=10\text{kN} \)
- The horizontal beam is chosen with a very large \( A \) to have the \( EA \) go to infinity.
Because the only interest is in the uh the structure is modelled with only the diagonals supported at the bottom as a pin and in the top only in z direction. For the rest all elements are modelled as bars, meaning they cannot take any moment.

According to the hand calculations the displacement would be:

\[ U_h = 0.227 \text{ mm} \]

GSA outputs:

![Figure B2](image)

Horizontal displacement, Print GSA, purple is force, green is axial force in elements and red is displacement.
B.2. **Vertical Displacement**

Here the vertical displacement due to a moment is calculated:

![Figure B3](image)

Figure B3
truss resisting a moment load via axial forces in the diagonals

\[ M = F_d B \]
\[ F_v = F_d \sin \theta \]
\[ F_d = EA \varepsilon \]
\[ \varepsilon = \frac{e_d}{L_d} \]
\[ e_d = u_v \sin \theta \]
\[ L_d = \frac{h}{\sin \theta} \]
\[ \Delta \beta = \frac{2u_v}{B} \]

This can be worked into the next formula:

\[ u_v = \frac{Mh}{BEA \sin^3 \theta} \]
In this test case:

- \( E = 205000 \text{ N/mm}^2 \)
- \( h=b= 5 \text{ m}, \) therefore \( \theta = 45^\circ \)
- the diagonals are chosen to be CHS 102*5.0
  - \( A = 1520 \text{ mm}^2 \)
  - \( I = 1770000 \text{ mm}^4 \)
- \( F_v = 10kN \) leading to a moment of 50kNm

The horizontal beam is chosen with a very large \( I \) to have the EI go to infinity.

Because the only interest is in the \( uv \) the structure is modelled with only the diagonals supported at the bottom as a pin and in the top only in \( z \) direction. The diagonal elements are modelled as bars, meaning they cannot take any moment, and the horizontal beam is modelled with infinite bending stiffness.

According to the hand calculations the displacement would be:
\( U_h = 0.4539 \text{ mm} \)

GSA outputs:
APENDIX C

PART OF THE DIAGRID
Part of the Diagrid

C.1. Deriving the formulae

The formulae described in the previous part will be slightly adapted to take into account the extra diagonals. Thereby the shear force, \( V \), and the bending moment, \( M \), are expressed in terms of relative displacement and rotational measures, \( \Delta u \) and \( \Delta \beta \).

\[
\begin{align*}
V &= K_\gamma \Delta u \\
M &= K_\beta \Delta \beta
\end{align*}
\]

With:

\[
\begin{align*}
\Delta u &= \gamma h \\
\Delta \beta &= \chi h \\
\Delta \beta &= \frac{2u_v}{B}
\end{align*}
\]

In these formulae \( \gamma \) is the shear factor, \( \chi \) is the bending factor or curvature and \( h \) is the height of the diagonals.

\[
h = L_d \sin \theta
\]
The force, \( V \), will be equally divided by \( N_w \) diagonal sets and therefore \( K_T \) could be written as:

\[
K_T = 2N_w A_{\text{web}} E \frac{\cos^2 \vartheta}{L_d}
\]

In this case \( N_w = 6 \).

The moment, \( M \), will be divided over the diagonals in the flange and therefore \( K_B \) could be written as:

\[
K_B = N_f B^2 A_{\text{flange}} E \frac{\sin^2 \vartheta}{L_d}
\]

In this case \( N_f = 6 + \) the additional help it gets from the diagonals in the web.

This leads to a \( N_f \) of roughly \((6 + 1/2) \times 2 + 1/2 = 10\)

When this is rewritten to a displacement the following formulae can be obtained;

The horizontal displacement due to the lateral force:

\[
u_{c,x} = \frac{V L_d}{2 N_w E A_{\text{web}} \cos^2 \vartheta}
\]

The vertical displacement due to the Moment:

\[
u_v = \frac{M L_d}{BN_f E A_{\text{flange}} \sin^2 \vartheta}
\]
Now if we consider the floor to be infinite bending stiff and the rotation in the connection to be 0 then via the uv a horizontal displacement can calculated, since then:

$$\Delta \beta = \frac{2u_v}{B}$$

This formula calculates the rotation that the structure makes due to in both the horizontal plane and the vertical plane thus:

$$u_h = h\Delta \beta$$

And from this follows:

$$u_h = \frac{2u_v h}{B}$$

And therefore uh due to bending can be calculated with:

$$u_{h,b} = \frac{2ML_dH}{B^2N_fEA_{flange} \sin^2 \vartheta}$$

Or with the simplified formula:

$$u(H) = \gamma H + \chi H^2$$

Now this can be filled out to:

$$u_{total} = \frac{VL_d}{2N_wEA_{web} \cos^2 \vartheta} + \frac{2ML_dH}{B^2N_fEA_{flange} \sin^2 \vartheta}$$

And since the beams are mainly loaded with axial force the tension can easily be calculated because if you know the loads from either the lateral load or bending the Fd can be derived via the same formulae as used with the braced portal frame:

$$F_h = 2F_d \cos \vartheta$$

And thus

$$F_d = \frac{F_h}{2 \cos \vartheta}$$
And with:

\[ F_d = A \sigma \]

\[ \sigma_{web} = \frac{F_h}{N_w A_{web} 2 \cos \theta} \]

For the tension in the flange can then be found:

\[ M = F_d B \]
\[ F_v = F_d \sin \theta \]

And thus:

\[ F_d = \frac{M}{B \sin \theta} \]

With:

\[ F_d = A \sigma \]

\[ \sigma_{flange} = \frac{M}{N_f A_{flange} B \sin \theta} \]
C.2. Validating the model

The first check is to see if the loads are equally distributed if the horizontal floor is modelled as infinitely stiff (both axial and bending) and the crosses are again modelled as bars so no moments will be taken by the diagonals.

In this test case:

- \( E = 205000 \text{ N/mm}^2 \)
- \( h = 21 \text{m} \)
- \( b = 3 \times 16.8 \text{m} \)
- \( \theta = 51.34^\circ \)
- the diagonals are chosen to be CHS 406 x 6.3
- \( A = 7920 \text{ mm}^2 \)
- \( I = 158500000 \text{ mm}^4 \)
- \( F_h = 2 \times 100 = 200 \text{kN} \)
- \( N_w = 2 \times 3 = 6 \)

With these values the calculated displacement according to the formulae should be:

\[
  u_{n,x} = \frac{VL_d}{2N_mE_{A_{web}} \cos^2 \theta}
\]

According to GSA these values are a bit larger being 0.7687. The reason for this could be that there is some redistribution in the diagonals because the floor is not really infinitely stiff. As can be seen in the GSA model, the two diagonals connected to the flange are loaded with 31.3 kN instead of 26.7 kN as the diagonal in the middle and in this case the lateral movement causes some tilting in the structure, due to bending from the lateral force. This causes some extra displacement as well.
In case of the stress in the members’ the following results are gained from the formula:

\[
\sigma_{web} = \frac{F_h}{N_{A}A_{web}2\cos\theta}
\]

\[\sigma_{web} = 3.37 \text{ n/mm}^2\]

According to GSA this is exactly the value for the diagonals in the middle. Again the ones at the side connected to the flange is slightly higher as can be expected.

If we check the same model against a moment applied to the structure:
In this test case:

- \(E = 205000 \text{ N/mm}^2\)
- \(h=21\text{m}\)
- \(b= 3 \times 16.8\text{m},\)
- \(\theta=51.34^\circ\)
- the diagonals are chosen to be CHS 406 x 6,3
  - \(A = 7920 \text{ mm}^2\)
  - \(I = 158500000 \text{ mm}^4\)
- \(M=2 \times 100 \times 50.4= 5040\text{ kN} \text{m}\)
- \(N_f \text{ (for the displacement in uv)} = 2 \times (3 + 2 \times 0.5 \text{ (Flange plus connected web diagonal)})\)
$u_y = \frac{ML_d}{BNf EA_{flange} \sin^2 \vartheta}$

$U_{v,\text{flange}} = 1.358 \text{ mm}$

And the results from GSA are 1.397 mm:

Figure C5
vertical displacement due to a moment

In case of the stress in the members' the following results are gained from the formula;

$\sigma_{\text{flange}} = \frac{M}{N_f A_{\text{flange}} B \sin \vartheta}$

$\sigma_{\text{flange}} = 8.08 \text{ n/mm}^2$

And GSA shows the following values $8.317 \text{ n/mm}^2$
Now we can clearly see that there are 5 diagonals on either side working for the moment. This means that Nf should be 5 instead of 4 and all the values from the formula are again lower than the values derived via GSA. Again this can be explained because of the assumption that the moment only created a vertical movement and not a horizontal movement as well.

The horizontal movement that we did expect would according to the formula be:

$$u_{h,b} = \frac{2ML^2H}{B^2N_f E_{\text{flange}} \sin \vartheta}$$

$U_{h,b} = 0.57\text{mm (with } N_f = 8\text{)}$

According to GSA this value is 0.58mm
APENDIX D

VALIDATING THE TOWER
Validating the tower

D.1. Checking the displacement

The tower that was studied consists of:

- 30 floors all 3500mm high.
- With the crosses spanning 6 stories.
- The loads that are applied are placed on the corners and in this case are 100 kN large.
- The floors are over dimensioned and now about 2m thick to have ‘infinite’ bending and axial strenght

This leads to the following Shear force and Moment:

![Figure D1](image1.png)

**Figure D1**
Example of the modelled diagrid tower in GSA.

![Figure D2](image2.png)

**Figure D2**
Moment-line and shear force line with 5 point-loads acting on a cantilevered beam
To calculate the displacement each floor is separately calculated
For shear this leads to:

\[
\begin{align*}
\delta_{k,s} &= \frac{VL_d}{2NwAE_{web}\cos^2\theta} \\
\delta_{k,b} &= \frac{2ML_dH}{B^2NfAE_{flange}\sin^2\theta}
\end{align*}
\]

First Floor
V = 1000 kN
M = 42000 kNm
Nf = 6
H = 105 m
Uh;\text{s} = 3.54 mm
Uh;b = 15.72 mm
Displacement at this level: 3.54 + 3.144 = 6.68 mm

Second Floor
V = 800 kN
M = 25200 kNm
Nf = 6
H = 84 m
Uh;\text{s} = 2.83 mm
Uh;b = 7.55 mm
Displacement at this level: 6.68 + 2.83 + 3.144 + 1.886 = 14.54 mm

Third Floor
V = 600 kN
M = 12600 kNm
Nf = 6
H = 63 m
Uh;\text{s} = 2.12 mm
Uh;b = 2.83 mm
Displacement at this level: 14.54 + 2.12 + 3.144 + 1.886 + 0.94 = 22.64 mm

Fourth Floor
V = 400 kN
M = 4200 kNm
Nf = 6
H = 63 m
Uh;\text{s} = 1.41 mm
Displacement at this level: $22.64 + 1.41 + 3.144 + 1.886 + 0.94 + 0.314 = 30.34$ mm

Fifth Floor
V = 200 kN
M = 0 kNm
Nf = 6
H = 63 m
Uh;s = 0.7 mm
Uh;b = 0 mm
Displacement at this level: $30.34 + 0.7 + 3.144 + 1.886 + 0.94 + 0.314 = 37.34$ mm

This leads to a total displacement at the top of 37.34 mm

If this is done with GSA these are the results:
D.2. Checking the stress

For the same structure the stress in the web and flange members can be calculated via the before mentioned formulae:

\[
\sigma_{web} = \frac{F_h}{N_w A_{web} 2 \cos \vartheta}
\]

\[
\sigma_{flange} = \frac{M}{N_f A_{flange} B \sin \vartheta}
\]

Now we use the same forces again to calculate the stress instead of the displacement.

Leading to a stress in the lowest members with \( F_h = 1000 \text{kN} \) in this case all the force will be taken by the diagonals in the web this leads to an \( N_w = 6 \):

If these values are used in the above mentioned formula the outcome is:

\( \sigma_{web} = 16.85 \text{ N/mm}^2 \)

This is again exactly the same value as with GSA in the middle diagonals.
The same lateral (wind loading) causes a moment as well, this moment has to be resisted by the flange members as described in the next formula:

\[ \sigma_{\text{flange}} = \frac{M}{N_f A_{\text{flange}} B \sin \theta} \]

Again the bottom of the tower is checked, this is where the stress will be the largest. At this location the moment is \( M = 42000 \text{ kNm} \) and all this has to be taken by the flange members and thus \( N_f = 2 \times 3 = 6 \), which results according to the formula:

\[ \sigma_{\text{web}} = 22.45 \text{ N/mm}^2 \]

In GSA the value is slightly lower with 21.61N/mm2.

The total overview of stress in the tower is displayed in the following figure showing that the tower would benefit largely when the section properties differ along the height of the tower. This would lead to a simple optimisation of the kilograms steel.
Figure D6
GSA results for stress in the flange diagonals due to the moment caused by wind load

Figure D7
overview of GSA results for stress in the web and flange diagonals due to wind load
Figure D8 comparing the stress values varying from 31.31 N/mm² to 31.81 N/mm².
D.3. Checking the assumptions

In the above calculations a few assumptions where made:

- The diagonals are bars instead of beams, to prevent local moments in the beams increasing the stress.
- The floors are heavily over dimensioned.

Here the GSA outputs are given for stress and deflection for the three different cases:

1. As in the assumed calculations
2. Beams instead of bars
3. Beams instead of bars and ‘normal’ floor slab dimensions

**Stress:** (fig D8)

From the stress comparison it seems that there is no real difference from these three situations. Although the stress is a bit higher in the most realistic case, this is mainly because here there is some additional moment due to bending of the beam and the rotation of the floors, but compared to the steel structure the 500mm thick concrete slabs (spanning 16m) are still working like a stiff diaphragm.

**Displacement:** (fig. D9):

From the displacement comparison it seems that there is no real difference from these three situations. Although the displacement at the top is a bit higher in the most realistic case, this is mainly because here there is some additional rotation due to bending of the beam and the rotation of the floors, but compared to the steel structure the 500mm thick concrete slabs (spanning 16m) are still working like a stiff diaphragm.

**Conclusion**

From the above examples it shows that the results gained by the formulae approximate the values in GSA. Therefore there will be continued with the result derived by GSA even with more complex geometry.
Figure D9 comparing the Displacement values varying from:

- Option 1: 35.50 mm
- Option 2: 35.48 mm
- Option 3: 36.86 mm
APENDIX E

GRADUATION COMMITTEE & PERSONALIA AUTHOR
Graduation committee

The graduation committee consisted of the following members:

Prof. dr. ir. J.C. Paul MBA [1]
e-mail: Joop.Paul@arup.com
tel: +31 (0)20 305 8500
affiliation: Arup Amsterdam
Delft University of Technology
Faculty of Architecture
Department of Building Technology
Section Structural Design

Ir. J.L. Coenders
e-mail: Jeroen.Coenders@arup.com
tel: +31 (0)20 305 8500
affiliation: Arup Amsterdam
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Design & Construction
Section Structural and Building Engineering

Ir. J.W. Welleman
e-mail: J.W.Welleman@tudelft.nl
tel: +31 (0)15 278 4856
affiliation: Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Design & Construction
Section Structural Mechanics

Dr. Ir. R.M.F. Stouffs
e-mail: R.M.F.Stouffs@tudelft.nl
tel: +31 (0)15 278 1295
affiliation: Delft University of Technology
Faculty of Architecture
Department of Building Technology
Section Design informatics
Personal Information author

Ir. T.K. Uijtenhaak
e-mail: tijluytenhaak@hotmail.com
tel: +31 (0)6 1418 7134
affiliation: Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Design & Construction
Section Structural Mechanics
Structural Design Lab
Arup Amsterdam