preface

foreword

In one and a half year – starting from August 2005 – the idea of creating a structural design optimisation tool grew to the realisation of a script, written in VBA and using AutoCAD 2007 as a visualisation tool, which has the ability to optimise the allocation of a variable number of columns in a structural transfer zone. An extended preliminary study is executed in (computational) structural optimisation – book two –, and the aspects and characteristics of transfer zones, modular dimensions, and structural grid layouts are discussed – book one. Based on the knowledge of optimisation methods gained from the preliminary study phase and the research stage, a design tool, intended to be used in the conceptual and preliminary design stage, is written, resulting in a VBA-script [for information on VBA, see Appendix A] of over 4000 lines. A test structure is presented in the addendum, including the determination of the optimal GA operator parameters.

This book is part of an integral report containing all findings of the Master's thesis on structural optimisation for the Delft University of Technology, Faculty of Civil Engineering and Geosciences, department of Building Engineering, and the Structural Design Lab (SDL). Two books including an addendum form the complete and final report of this Master's thesis.

book one:	**transfer zones in multi-use buildings and the development and analysis of the optimisation tool**
book two:	**background studies on structural optimisation**
addendum:	**determination of the optimal GA operator parameters and the presentation of test results based on example structures**

The reader of the complete report will find that the emphasis is mainly on the study into genetic algorithms and the onset of the design optimisation tool.

[It needs to be mentioned that in this Master's thesis the adjective 'structural' will be used to name aspects considering building engineering and structural engineering matters, rather than structural aspects in general]

Delft, January 2007

R.J. (Roel) van de Straat

graduation committee

name: **prof. dipl.-ing. J.N.J.A. Vamberský** – chairman of the committee
university: Delft University of Technology
Faculty of Civil Engineering and Geosciences
address: Stevinweg 1
2628 CN Delft, room 1.35 St-II
The Netherlands
telephone: +31152783174 (secretariat)
e-mail: j.n.j.a.vambersky@citg.tudelft.nl

name: **ir. J.L. Coenders** – daily supervisor
university: Delft University of Technology
Faculty of Civil Engineering and Geosciences
address: Stevinweg 1
2628 CN Delft, room 1.58 St-II
The Netherlands
telephone: +31152785711
e-mail: j.l.coenders@citg.tudelft.nl

name: **ir. J.W. Welleman**
university: Delft University of Technology
Faculty of Civil Engineering and Geosciences
address: Stevinweg 1
2628 CN Delft, room 6.15
The Netherlands
telephone: +31152784856
e-mail: j.w.welleman@citg.tudelft.nl

name: **ir. S. Boer**
organisation: ONL [Oosterhuis_Lénárd] (till March 2006) and Mecanoo architecten
home address: Goudsesingel 464
3011 KP Rotterdam
telephone: +31648082965
e-mail: sanderboer@myrealbox.com

consultant

name: **dipl. inform. F. Scheurer**
university: Eidgenössische Technische Hochschule Zürich
Faculty of Architecture, Chair of CAAD
address: Hoenggerberg HIL E15.1
CH-8093 Zürich
telephone: +41446334025
e-mail: scheurer@arch.ethz.ch

acknowledgements

Many people helped in the realisation of this Master's thesis, and it is therefore underlined that, without their input and effort, the challenge of writing this report would be far greater. That is the reason why here those people are thanked wholeheartedly.

The graduation committee
in general is thanked for the positive input, the advice, and the criticism given as a group. Unanimity, consistency, and perseverance are characteristics the committee cannot be denied.
In particular, **prof. dipl.-ing. J.N.J.A. Vamberský**, for his ability to give an answer to a question just by listening, his great passion for the building engineering practise (and the way this is transferred to his students), and the eagerness to comprehend new techniques and methods shown by his students. **Ir. J.W. Welleman**, for his enormous ardency and enthusiasm, especially after having spent a considerable amount of time at home due to illness. Extensive collaboration on the Discrete Element Method led to the implementation of this method in this Master's thesis. **Ir. J.L. Coenders**, for the introduction to and shared knowledge of computational design and structural optimisation. Due to his work for the Structural Design Lab the addition of the name of this group on the front page of this Master's thesis, really means something in the computational and building engineering practise. **Ir. S. Boer**, as the only committee member not associated with the Delft University of Technology, for dealing with the extra time he put into this project, and for his insights in the architectural practise. Something not only used for this Master's thesis, but which will also be used for the upcoming Master's study Architecture.

Dipl. inform. F. Scheurer
is willingly thanked for being the non-official member of the graduation committee and for sharing his knowledge on computational design and especially on genetic algorithms. Based on his papers, and visits to Delft, better insights were received into this complex matter.

Ir. A. Habraken
of Arup Amsterdam who was open for an invitation from him to the Arup office in Amsterdam and could answer various structural related questions on the Groningen Twister project, and thereby supplied better understanding in the design of column supported slabs.

ir. S. de Groot and ir. E. Quispel
for their preliminary insights on the subject of stacking different functional layouts within a multi-use building and their willingness to share their work.

H. Ohmori, prof. dr. J. Strasky, and ir. R. Uytenhaak
who supplied images and text files after requests via e-mail.

ir. R. Pijpers
who, with enormous patience, was able to install AutoCAD 2007 on the desktop at the workspace, which was of essential importance for finishing this Master's thesis.

the graduation students of room 0.72
for their tips and tricks on how to graduate, the laughs that were shared during the complete Master's thesis period, and for the many final presentations and accompanying drinks from those who already graduated.

the fellow board members and friends of the Stichting Dispuut Utiliteitsbouw
who made the finishing of the 'U-profiel' of May 2006 a lot more bearable, who opened up the continent of Asia for an unforgettable study tour to Dubai and the east of China at end of the winter of 2006, where, visiting Shanghai, the research stage of this Master's thesis already could be presented to Chinese fellow students. And of course for their friendship, during the whole combined period of working on the Master's thesis and the board membership.

my parents, my brother, and his girlfriend
for being the biggest support, a graduate student could wish for. Enough said.

And of course everyone else, who, although not contributing directly to the content of this Master's thesis, showed that there is more to life than columns, structural grids, and genetic algorithms, but who contributed to the social and emotional evolution of the author.

summary

general introduction

As more functional requirements are implemented in one building, more pressure on the structural designers, both architects and building engineers, will be transferred. Especially in dense urban areas, multi-use buildings usually have several different layouts, for instance for an underground parking, a ground floor, office floors, and mechanical floors. One of the main problems here is the transfer of loads between the different structural grid systems of the different functions, and as a result, numerous structures are built and will be built with inefficient transfer zones considering the volume of material needed for a floor or the amount of reinforcement or prestressing needed. In lieu of making compromises in the design of structural grid systems per function in order to vertically align the grid lines or grid bands, a structural transfer zone in an intervening floor can be designed. By designing an optimal allocation of vertical oriented structural elements, loads can be transferred between the different structural grid systems, without adversely affecting the functionality and usability of the intervening floor.

structural optimisation design tool

The Master's thesis 'Optimisation of structural transfer zones in multi-use buildings' deals with the development of a computational structural design tool based on an artificial intelligence method, that can determine the optimal solution for the allocation of columns between two structural grid systems. Based on genetic algorithms as an optimisation technique and by using basic and simple rules, the design tool can be used for the determination of size, angle, and placement of load bearing columns of the intervening floor. This tool, implemented in VBA and using AutoCAD as a visualisation tool, allows the user to generate and optimise the configuration of the load bearing elements for an arbitrary design, with the rules following from the demands of several aspects of a structural and functional design. With the addition of non-structural criteria (such as costs, aesthetics and construction) the engineer can complete the design, and create an optimised design that is based on the integration of the load bearing elements into a functionally efficient building floor, rather than being only based on the stress-weight optimisation of the individual components.

features of the tool

In the AutoCAD environment, the user needs to give the starting and end point of the structural grid line graphically in both the bottom and top layer of the intervening floor to determine the possible location of the columns. Subsequently, the vertical point load, the horizontal point loads, and the moments in the centre of gravity acting on the load transferring structure need to be given numerically. Based on the user input, the tool then randomly generates several solutions in what is called the first generation. By determining the fitness, or in other words the overall compliance with the prescribed desired conditions, the genetic algorithm will use the best solutions to create a new population. This process, including several other genetic algorithm operators, will be repeated until the fittest or best solution per population remains unaltered during a number of generations.

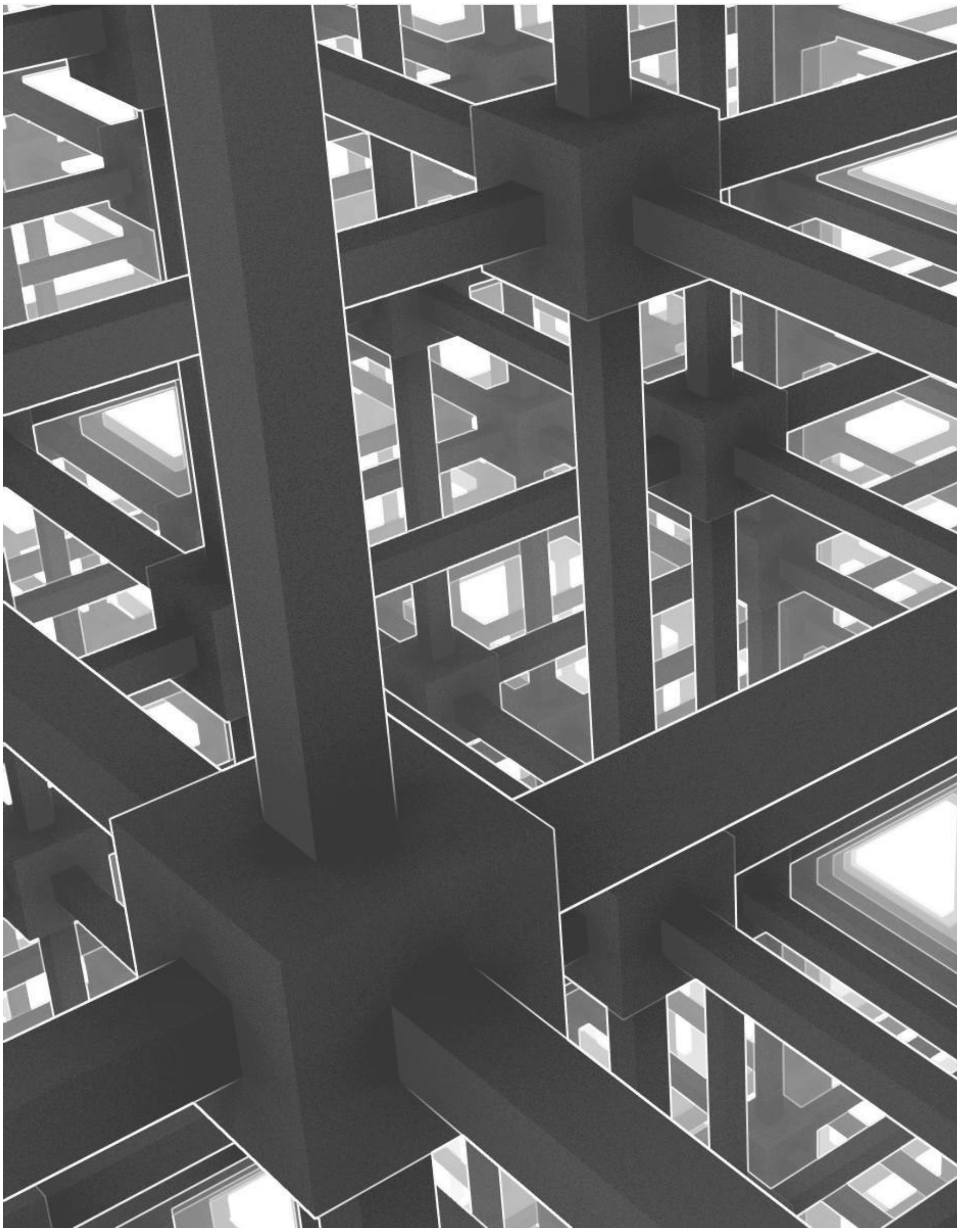
concluding remarks

This Master's thesis shows the capability of an artificial intelligence based design optimisation tool in a predefined setting of the allocation problem of columns in a structural transfer zone. At the same moment, it is made clear that progress can be made for the presented design tool and in scripting design tools for the building practise in general. This also means that it can be expected that tools similar to the tool presented in the Master's thesis will be used more often in the near future. This, however, does not mean that the structural engineer will lose his or her position, as hand calculations and logical interpretations of the result of the design tools will always have to be made.

table of contents

references

appendices



Cubic Space Division, M.C. Escher, litho, 1925, 27 x 26,5 cm[1]

One another intersecting rectangular beams divide each other in pieces of equal length, which each are the edge of a cube. So, the space is filled to infinity with cubes of the same volume

---

[1] Escher [6]

1,0 introduction

1 introduction

general introduction to the Master's thesis project

The *manual optimisation* and iteration to minimise weight, cost or for ease of construction issues of structures are instances of non-computational optimisation that are widely used by structural engineers. This includes physical modelling, with the hanging models of Antonio Gaudí as a striking example [Figure 1.1]. By making a physical (scale) model and subjecting it to internal and external loads, a designer can learn about the internal forces in the structural elements and remodel the design accordingly and thus working towards an optimum design. Therefore, when using an optimisation technique like physical modelling, an engineer is able to adapt the initial design in such a way that it will meet the non-structural boundary conditions, while corresponding to e.g. compression lines or minimal energy surfaces.

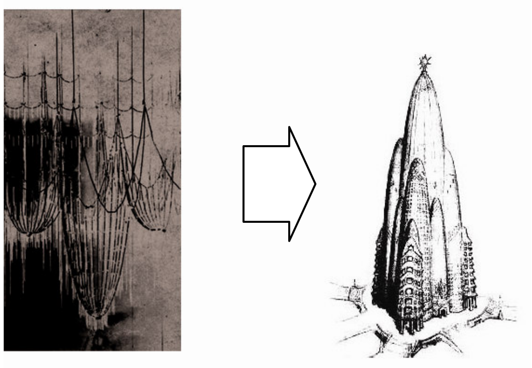


Fig. 1.1. Hanging model by Gaudí [www.gaudiclub.com, August 2005] and a drawing of Hotel New York, design by Gaudí [www.vitruvius.com.br, August 2005] (Model and drawing are not of the same project)

One of the advantages of physical modelling is the insight in the behaviour of the modelled structure[2]. The same can hold true for analysing a virtual model after subjecting it to a *computational* analysis, but often the output will give the user a

---

[2] Coenders [4]

list containing data without revealing how this data was obtained. But with two disadvantages of physical modelling being two advantages of computational methods (viz. the amount of time to make a model and the simplicity of adjusting the model), together with the accuracy of the latter mentioned, engineers need to comprehend the modelling of a structure based on computational optimisation techniques.

Many *computational optimisation techniques*, based on both 'classical' and 'artificial intelligence' methods can be used by structural engineers in all kinds of fields as well. Nevertheless, for building design only very basic design tools exist based on optimisation. Usually these tools only include a stress versus weight trade-off. One field where an implementation of optimisation-based design tools can give good and maybe surprising results is in the allocation of structural elements in the structural transfer zones of buildings between different layers of functional program and routing. One example of this class of optimisation problems is shown in the Groningen Twister project[3]. A similar technique that was used for this project can be used for the determination of size, angle, and placement of load bearing columns of an intervening floor, such as a ground floor or a technical floor in a high-rise building. The main reason for using computational methods in dealing with this structural problem is the difficulty that comes with the different structural grids of the different building floors, and that a logical allocation of the load bearing elements might not be evident.

formulation of the general objectives

The goal of the final report is to demonstrate the use of optimisation algorithms in building design for combinations of structural and non-structural criteria as well as the usability of optimisation techniques for combined architectural and structural engineering purposes. This, in particular for the allocation problem of load bearing structural elements of intervening floors as an initiation of the structural design phase, with the use of genetic algorithms as an optimisation technique.

The demonstration will result in the evaluation of a structural design tool that allows the user to generate and optimise the configuration of the load bearing elements, with the rules following from the demands of several aspects of a structural and functional design. With the addition of non-structural criteria (such as costs, aesthetics and construction) the engineer can complete the design, and create an optimised design that is based on the integration of the load bearing elements into a functionally efficient building floor, rather than being only based on the stress-weight optimisation of the individual components.

---

[3] Scheurer [19]

composition of the report

book one – transfer zones in multi-use buildings and the development and analysis of the optimisation tool

Book one of the Master's thesis deals with the recognition of the problem and a possible solution in the form of a design optimisation tool. As in multi-use buildings, different functions and therefore different structural grid systems are present difficulties can occur when these systems need to be connected. The necessity of transfer zones will be discussed in Chapter 2, along with summaries of two Master's theses of Building Engineering graduates. Chapter 3 deals with the dimensions and different considerations concerning structural grid layout in multi-use buildings. Main chapter of book one is Chapter 4. Here the different characteristics and design aspects of the optimisation tool will be illuminated. In Chapter 5 the validation of the tool will be given on the basis of test runs.

book two – background studies on structural optimisation

The second book of the Master's thesis discusses the concepts of structural optimisation in building engineering, including an enumeration of optimisation methods and an in-depth research on genetic algorithms. In Chapter 2, an introduction into structural optimisation is given and the philosophy behind it is discussed. Chapter 3 and 4 form the main chapters of book two. Here, the optimisation methods and in particular the new artificial intelligence▹ method genetic algorithms are dealt with. This book concludes with building projects where optimisation played a key role.

To conclude the introduction Q.Q. Liang [10] can be quoted by stating that the "challenge in structural optimisation is to transform it from an exotic and fruitless academic exercise into a rational and efficient design tool for practicing building engineers". The work presented in this Master's thesis is to answer this challenge.

Fig. 1.2. 'Columns' on a predefined grid [www.uni-kl.de, September 2005]

▹ Traditional artificial intelligence has been centered around the idea of representation of the world. Toward the end of the 1980s, an exciting new field appeared in computational intelligence; 'embodied cognitive science', also known as 'new artificial intelligence'. It was suggested that the discussion about thinking, logic, and problem solving was based on assumptions that come from our own introspection, from how we tend to see ourselves. Next, it was suggested that these assumptions need to be dropped, that we so away with thinking and with what people call high-level cognition and focus on the interaction with the real world. Intelligence must have body, hence 'embodied intelligence'.
From 'Understanding Intelligence' by R. Pfeifer [15]

2 transfer zones in multi-use buildings

In this chapter, the necessity of transfer zones [Section 2.1] is shortly illustrated, including the possible problems that have to be dealt with when designing these structures. In Section 2.2 the contents of two Master's theses of former Building Engineering students from the Delft University of Technology, (partially) dealing with this same subject, are summarised.

2.1 necessity of transfer zones in multi-use buildings

In this Master's thesis, a transfer zone is the structure or a combination of structural elements that transfers loads between two structural floors in a building. When dealing with utility buildings, - especially those with multiple functions - usually different structural grid layouts are needed to optimise the functionality of the different floors and spaces of these buildings [see also Chapter 3]. Consider for instance the different functional layouts that are needed for an underground parking, a ground floor, an office floor, and a mechanical floor in just one building. A possible problem that can occur is that those different structural grids are not compatible with each other, in a sense that the grid points or grid bands are not vertical aligned over the different floors of the concerning building [Figure 2.1].

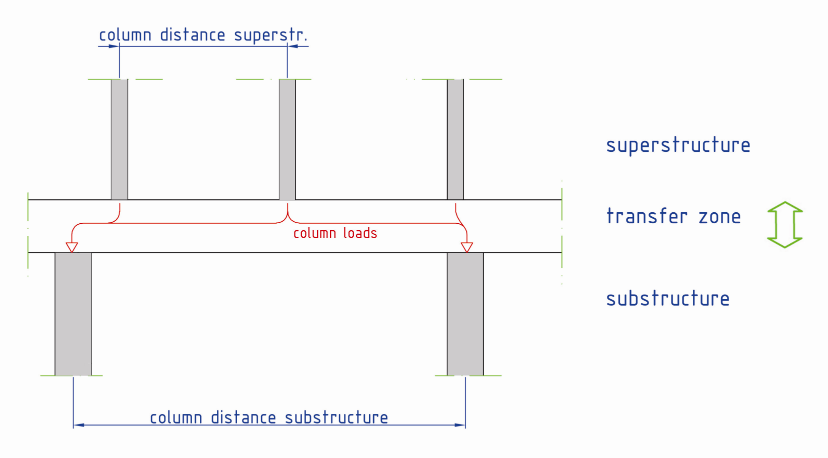


Fig. 2.1. Variation in structural grids in a building

As a result, inefficient transfer zones have to be designed. Inefficient in a sense that these zones require relatively much space. In some cases, a thick concrete slab is used to transfer the loads from one floor to its underlying floor. When high loads need to be transferred and the vertical alignment of the structural grids of the different floors is poor, the thickness of this concrete slab can easily become over 1 metre, including prestressing bars. Especially for foundation slabs, this thickness is not uncommon.

To reduce the thickness of the transfer zone, the different structural grids could be aligned with each other as good as possible. However, making a compromise by altering the layout of both floors will lead to inefficient structures or functional layouts on both sides of the transfer zone. In general, the building function with the highest profit per square metre of the lettable area is of the biggest influence of the used modular dimensions. Usually, this is the commercial area of a building. The adaptation of the functional layout of an underground parking to the layout of the commercial area could decrease the functionality of the underground parking is such a way that the lettability would fall below a profitable situation.

To overcome the difficulties of altering the structural or functional layout and/or construction of a relatively thick concrete slab, the super- and substructure can be pulled apart from each other and an intervening (column) structure can be designed to transfer the loads, which will be discussed in this Master's thesis.

2.2 Master's theses on the subject of transfer zones

In this section, two Master's theses [2.2.1 and 2.2.2] on the subject of transfer zones will be summarised. The first thesis from Sierk de Groot[4] (who graduated from the faculty of Civil Engineering and Geosciences at the Delft University of Technology in December 2003) deals with the allocation of slanting columns between an underground parking and a residential building. The second thesis from Elsbeth Quispel[5] (who graduated from the faculty of Civil Engineering and Geosciences at the Delft University of Technology in September 2003) deals with the stacking of functions on top of underground parkings.

2.2.1 Master's thesis *'De Componist'* – S. de Groot

introduction of the Master's thesis

After plans were developed to replace residences on the Carmenlaan in the city of Amstelveen, The Netherlands with a new housing block, ONX Architects was invited

---

[4] De Groot [7]

[5] Quispel [16]

to make the architectural design[6]. This resulted in the design of 'The Componist', a residential building with 123 apartments, an underground parking and a day-care centre. The main characteristics of the building are its length of approximately 200 metre and the lift of the building of 5,8 metre above ground level [Figure 2.2].

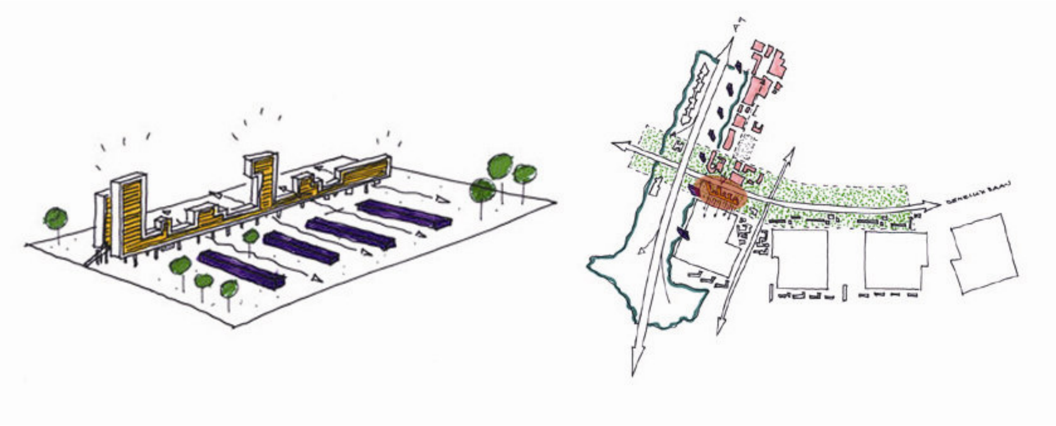


Fig. 2.2. A 3D-sketch of 'The Componist' and a top view of the building [7]

The lift is realised by placing the building on slanting columns. The angle of the columns results from the structural connection between the different grid lines of the residential function, floating above the ground level and the underground parking [Figure 2.3]. The functional arrangement demands that the structural elements, the columns, for both functions are placed on the grid lines.

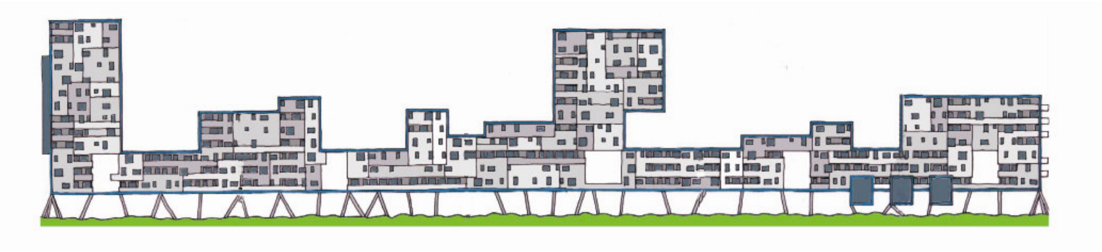


Fig. 2.3. Side view of 'The Componist' including the slanting columns [7]

---

[6] ONX Architecten bna [25]

elaboration of the Master's thesis

During his Master's thesis Sierk de Groot made a study into the structural possibilities of the slanting columns. In this study, three boundary conditions for the final allocation of the columns are given.

1. the columns need to be placed under an angle, as was intended by the architect;
2. the columns have to be capable to transfer the loads from the building to the underground parking / foundation [Figure 2.4 on the next page];
3. the columns have to make sure that the stability of the total structure is guaranteed and that the deformations are within certain boundaries.

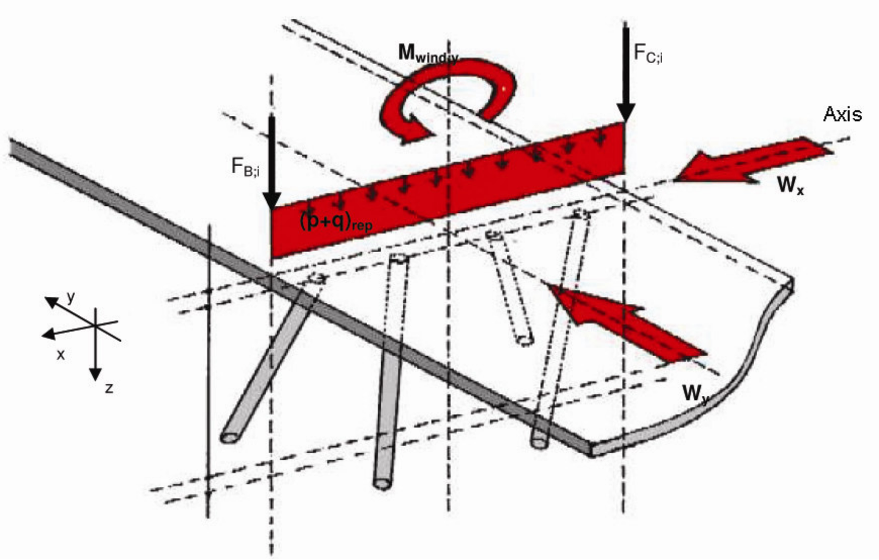
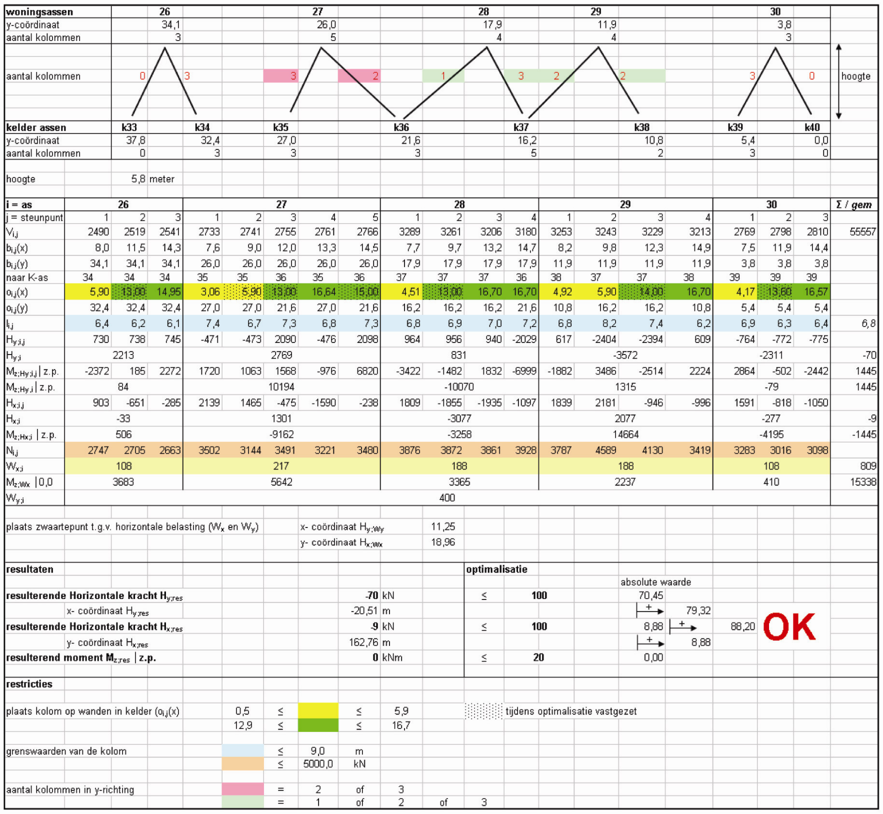


Fig. 2.4. Forces acting on the structure in different directions [7]

As stated before, the hinged columns cannot be placed randomly. The possible locations of the columns are defined by the grid lines of the residential building and the underground parking. To reduce complexity, the columns connect the grid lines of the residential building with the nearest grid lines of the underground parking, where the number of columns per axis may vary.

The horizontal loads, as a result of the angle of the columns not being equal to 90°, are transmitted by the floor slab of the residential building and the parking deck. No comments are given, concerning the ability of these floors to transfer the (horizontal) loads.

With a 'hand calculation', the location of the columns was determined and the values of the forces acting on them were estimated. First, the number of columns was determined, including the location of the top of the columns under the floor slab of the building, where the columns are set 500 mm off the edges of the walls (at the façade) that are on the grid lines of the residential building. The number of columns is dependant on the permanent load and a maximum load bearing capacity of the columns. Second, the distance in the y-direction from the top to the bottom of the columns was determined. This decision was made based on the minimal resulting horizontal forces in the y-direction, which was found by using a MS Excel sheet [Figure 2.5]. Finally, the distance in the x-direction, so with a constant y-coordinate, was determined. This again based on the minimal resulting horizontal forces, only this time in the x-direction [Figure 2.6, next page].



woningsassen	**26**	**27**	**28**	**29**	**30**
y-coördinaat	34,1	26,0	17,9	11,9	3,8
aantal kolommen	3	5	4	4	3

aantal kolommen	0	3	3	2	1	3	2	2	3	0

kelder assen	**k33**	**k34**	**k35**	**k36**	**k37**	**k38**	**k39**	**k40**
y-coördinaat	37,8	32,4	27,0	21,6	16,2	10,8	5,4	0,0
aantal kolommen	0	3	3	3	5	2	3	0

hoogte 5,8 meter

i = as	**26**			**27**					**28**				**29**				**30**			***Σ / gem***
j = steunpunt	1	2	3	1	2	3	4	5	1	2	3	4	1	2	3	4	1	2	3	
$V_{i,j}$	2490	2519	2541	2733	2741	2755	2761	2766	3289	3261	3206	3180	3253	3243	3229	3213	2769	2798	2810	55557
$b_{i,j}(x)$	8,0	11,5	14,3	7,6	9,0	12,0	13,3	14,5	7,7	9,7	13,2	14,7	8,2	9,8	12,3	14,9	7,5	11,9	14,4	
$b_{i,j}(y)$	34,1	34,1	34,1	26,0	26,0	26,0	26,0	26,0	17,9	17,9	17,9	17,9	11,9	11,9	11,9	11,9	3,8	3,8	3,8	
naar K-as	34	34	34	35	35	36	35	36	37	37	37	36	38	37	37	38	39	39	39	
$o_{i,j}(x)$	5,90	13,00	14,95	3,06	5,90	13,00	16,64	15,00	4,51	13,00	16,70	16,70	4,92	5,90	14,00	16,70	4,17	13,60	16,57	
$o_{i,j}(y)$	32,4	32,4	32,4	27,0	27,0	21,6	27,0	21,6	16,2	16,2	16,2	21,6	10,8	16,2	16,2	10,8	5,4	5,4	5,4	
$l_{i,j}$	6,4	6,2	6,1	7,4	6,7	7,3	6,8	7,3	6,8	6,9	7,0	7,2	6,8	8,2	7,4	6,2	6,9	6,3	6,4	*6,8*
$H_{y;i,j}$	730	738	745	-471	-473	2090	-476	2098	964	956	940	-2029	617	-2404	-2394	609	-764	-772	-775	
$H_{y;i}$	2213			2769					831				-3572				-2311			-70
$M_{z;Hy;i,j}$ | z.p.	-2372	185	2272	1720	1063	1568	-976	6820	-3422	-1482	1832	-6999	-1882	3486	-2514	2224	2864	-502	-2442	1445
$M_{z;Hy;i}$ | z.p.	84			10194					-10070				1315				-79			1445
$H_{x;i,j}$	903	-651	-285	2139	1465	-475	-1590	-238	1809	-1855	-1935	-1097	1839	2181	-946	-996	1591	-818	-1050	
$H_{x;i}$	-33			1301					-3077				2077				-277			-9
$M_{z;Hx;i}$ | z.p.	506			-9162					-3258				14664				-4195			-1445
$N_{i,j}$	2747	2705	2663	3502	3144	3491	3221	3480	3876	3872	3861	3928	3787	4589	4130	3419	3283	3016	3098	
$W_{x;i}$	108			217					188				188				108			809
$M_{z;Wx}$ |0,0	3683			5642					3365				2237				410			15338
$W_{y;i}$	400																			

plaats zwaartepunt t.g.v. horizontale belasting (W_x en W_y)	x- coördinaat $H_{y;Wy}$	11,25	
	y- coördinaat $H_{x;Wx}$	18,96	

resultaten			**optimalisatie**					
					absolute waarde			
resulterende Horizontale kracht $H_{y;res}$	**-70**	kN	≤	**100**	70,45			
x- coördinaat $H_{y;res}$	-20,51	m			$\xrightarrow{+}$	79,32		
resulterende Horizontale kracht $H_{x;res}$	**-9**	kN	≤	**100**	8,88	$\xrightarrow{+}$	88,20	**OK**
y- coördinaat $H_{x;res}$	162,76	m			$\xrightarrow{+}$	8,88		
resulterend moment $M_{z;res}$ | z.p.	**0**	kNm	≤	**20**	0,00			

restricties									
plaats kolom op wanden in kelder ($o_{i,j}(x)$)	0,5	≤	🟨	≤	5,9		▨	tijdens optimalisatie vastgezet
	12,9	≤	🟩	≤	16,7			
grenswaarden van de kolom	🟦	≤	9,0	m				
	🟧	≤	5000,0	kN				
aantal kolommen in y-richting	🟥	=	2	of	3			
	🟩	=	1	of	2	of	3	

Fig. 2.5. MS Excel sheet for the determination of the optimal location for slanting columns for 'The Componist' [7]

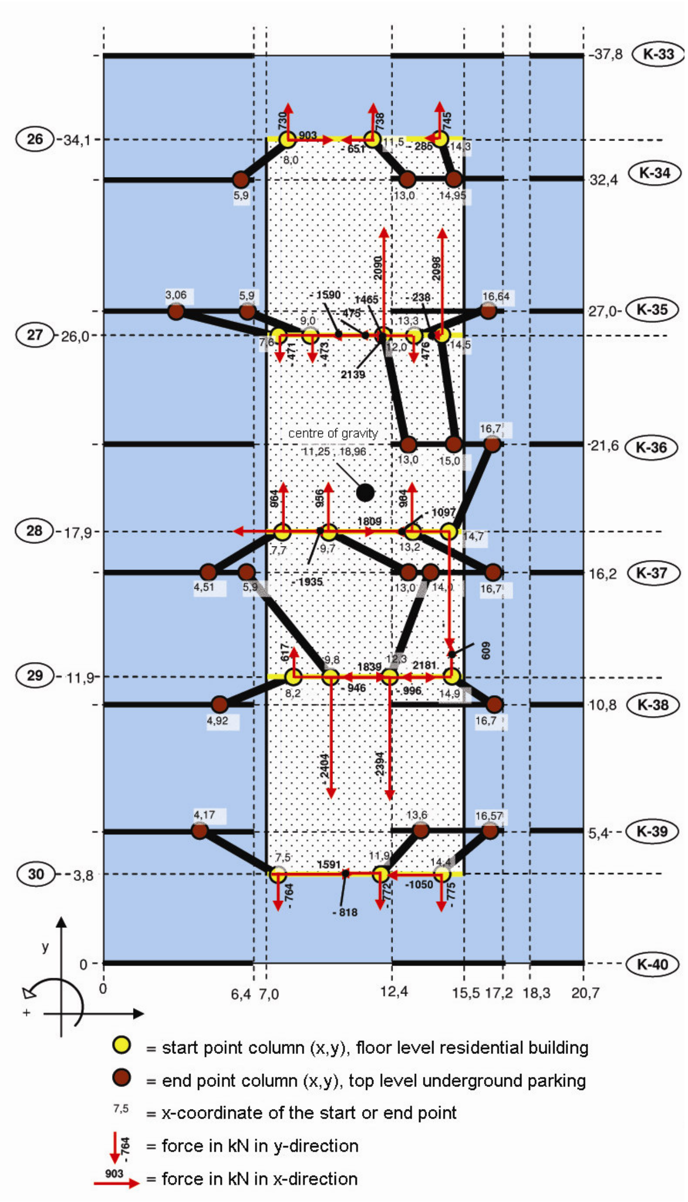


Fig. 2.6. Visualisation of the location of the start points and endpoints of the columns including the forces acting on them [7]

This 'hand calculation' was followed by a more extensive calculation in ESA-Prima Win to check the outcome of the allocation problem of the columns and to show that the 'optimal' allocation that was found with the MS Excel sheet was better than a randomly chosen configuration of columns.

2.2.2 Master's thesis '*Functiestapeling boven ondergrondse parkeergarages*' – E. Quispel

introduction of the Master's Thesis

In order to keep buildings accessible for people coming by car, sufficient parking facilities should be implemented in the design of these buildings. In combination with the lack of space, especially in the urban area, this often results the integration and the stacking of different building functions, and thus an increase in complexity of the design and in costs. The requirements of the superstructure hereby are often considered more important because of the higher value per square metre. This may lead to a decrease in the functionality of the (underground) parking, as the bearing structure of the building complicates the optimal arrangement of the parking spaces.

A solution for this problem could be the optimisation of the allocation of load bearing columns in the intervening floor, between the sub- and superstructure, so that a costly transition area could be unnecessary. If this would result in a satisfactory design, this principle could be adapted for other combinations of integration of functions, even for multiple intervening floors in one building, including the different structural grids of parking, office, residential and commercial floors. Obviously, the shape of the building or the different building sections, next to the functional arrangement is of great importance as well.

Quispel states in her formulation of the problem, that "the lack of overview over every design aspect of interest, including the accompanying points of attention and the restrictions, and the lack of understanding their influence during the design phase, leads to complicated structural solutions and/or inefficient functional arrangements in the design of underground parkings when dealing with stacking of functions". She claims that the following design aspects including their sub aspects should be mentioned for the design of an underground parking as a part of functional stacking.

- functional layout
    - positioning of the cars and routing (traffic circulation system)
    - location of the structural elements
    - shape of the parking
    - capacity, possibilities for expansion and future developments
    - demanded level of quality
    - access to the parking area (access ramps)
    - pedestrian facilities for horizontal and vertical transport
- main load bearing structure
    - downward and upward loading
    - location of the structural elements
    - floor types and floor spans

- building site applications and foundation
    - construction method and anchorage
    - excavation depth and excavation system
    - vertical boundary
- safety, social security and user-friendliness
    - traffic safety
    - fire safety
    - social security (convenient layout, accessibility, attractiveness)
    - routing

Modular dimensions for (underground) parkings

The quality of an underground parking depends for a large amount on the functional layout and this largely depends on the modular dimensions of this building type. Below some standard modular dimensions for parkings in The Netherlands are given. The three main aspects for the functional design of parkings are the positioning of the parked cars, the routing and the allocation of the columns and these three aspects depend on the modular dimensions of the parking. According to the Dutch standard NEN 2443 *Off-street and multi-storey car parks*[7], different dimensions are applicable for the width of the circulation roads, the width of the parking spaces, the depth of the parking spaces, etc. [Figure 2.7 and Table 2.1]

[for more on the modular dimensions and the structural grids of buildings, see Chapter 4]

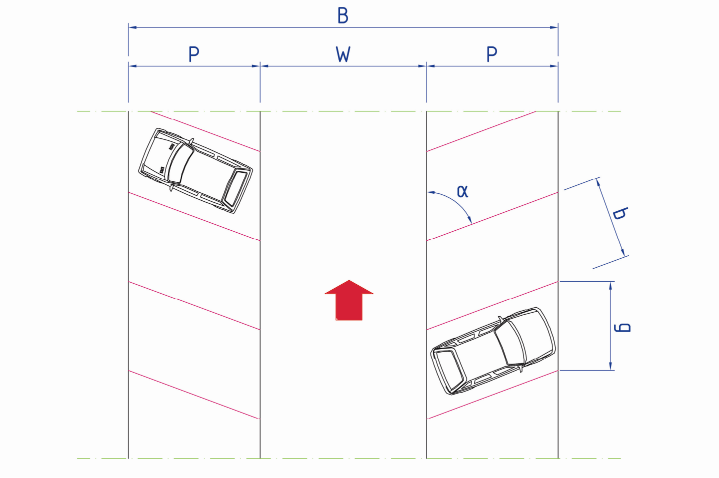


Fig. 2.7. Modular dimensions of a parking in case of a two-sided arrangement with a single parking strip (no parkings back to back) [16]

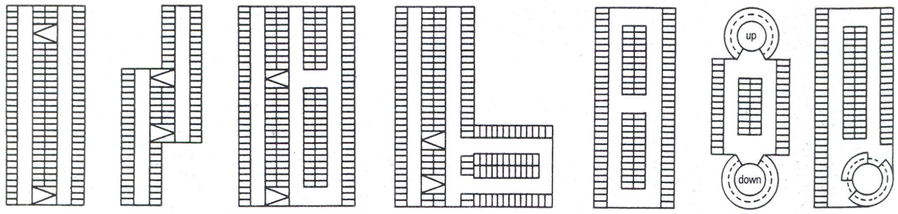
---

[7] Nederlandse norm NEN 2443 [13]

aspect of dimension		range of values [m]	recommendation
B - length parking unit	one-sided	8.15 - 11.35	
	two-sided	12.30 - 16.35	
P - depth parking unit	single parking strip	4.15 - 5.20	
	double parking strips	6.50 - 10.00	
W - width circulation road		4.00 - 6.35	
b - width parking unit		2.30 - 2.50	b_{min} = 2.35 m
g - parking width parking unit		2.30 - 5.00	
α - parking angle		30° - 90°	between 60° - 90°

Table 2.1. Applicable modular dimensions for parkings

Before tuning the layouts of the sub- and superstructure it is of importance to tune the different functions in the superstructure with each other. For parkings, the main load bearing structure can have two main directions; the direction of the parking spaces and the direction of traffic and usually consists of load bearing walls on the periphery of the parking, and interior columns. The allocation of the columns is determined by the structural grid or vice versa, depending on the viewpoint. However, nowadays and almost standard, the future tenant or owner of an underground parking demands a column-free parking area. However, this leads to an increase in complicity, especially when other functions need to be stacked on the parking area.



four rows

two offset sets of two rows

six rows

right-angled layout ramps in direction of traffic

multi-storey car -park with ramps

spiral ramps

Fig. 2.8. Plan views and arrangement of ramps [14]

2.3 concluding remarks

The Master's theses of both De Groot and Quispel deal with the stacking of different functional floors in multi-use buildings. De Groot claimed that a transfer zone of slanting columns between an underground parking and a residential building is possible and determined a possible column structure for one building block. Quispel even concluded that the optimisation of the allocation of load bearing columns between a super- and substructure could be a solution for the problem of possible concessions that have to be made when designing structural and functional layouts in multi-use buildings.

For this Master's thesis, the experiences of De Groot and Quispel are used, with the calculations on 'De Componist' as main point of interest. The foremost difference in the manner of calculation is that in this Master's thesis the determination of the optimal transfer zone is aimed for with an actual optimisation method, whereas De Groot used a MS Excel worksheet with which the designed structure could be checked. Another difference lies in the modelling of the super- and substructure. In calculating the column structure of 'De Componist' a non-rigid superstructure is used, including the determination of loads per grid lines. Although, a similar onset was intended for implementation in the VBA script, another route was finally chosen. The development and analysis of the design optimisation tool is given in Chapter 4 and 5.

3 modular dimensions and structural grid layouts

A short and general introduction into the modular dimensions of buildings and the terminology concerning structural grid layouts are given in Section 3.1. Section 3.2 deals with the modular dimensions of different building types and these are translated into structural grids in Section 3.3.

3.1 introduction and terminology

Studying existing buildings reveals that there are often strong and easily identifiable patterns present in the way buildings are functionally organised, and in the structural systems they use [Figure 3.1 on the next page]. The patterns formed by the structural configurations are related to those of the functional organisations. Each affects the other. To a large extent, the adoption of a particular functional configuration for a building often determines the structure used, and vice versa. For this reason, the process of designing a structure is implicitly linked to the process of designing the overall building. Neither process can really exist without the other[8].

It should be clear that these patterns, or structural grids, function as an aid for the architect and the structural designer. Besides, the sizes of many objects in a building are adapted to patterns of modular dimensions; office furniture, for instance. But apart from this, does the building industry still needs these structural grids? The shapes of buildings are becoming more and more free-formed and traditional structural elements are replaced by new ones [Figure 3.2, page 2,7].

---

[8] Schodek [11]

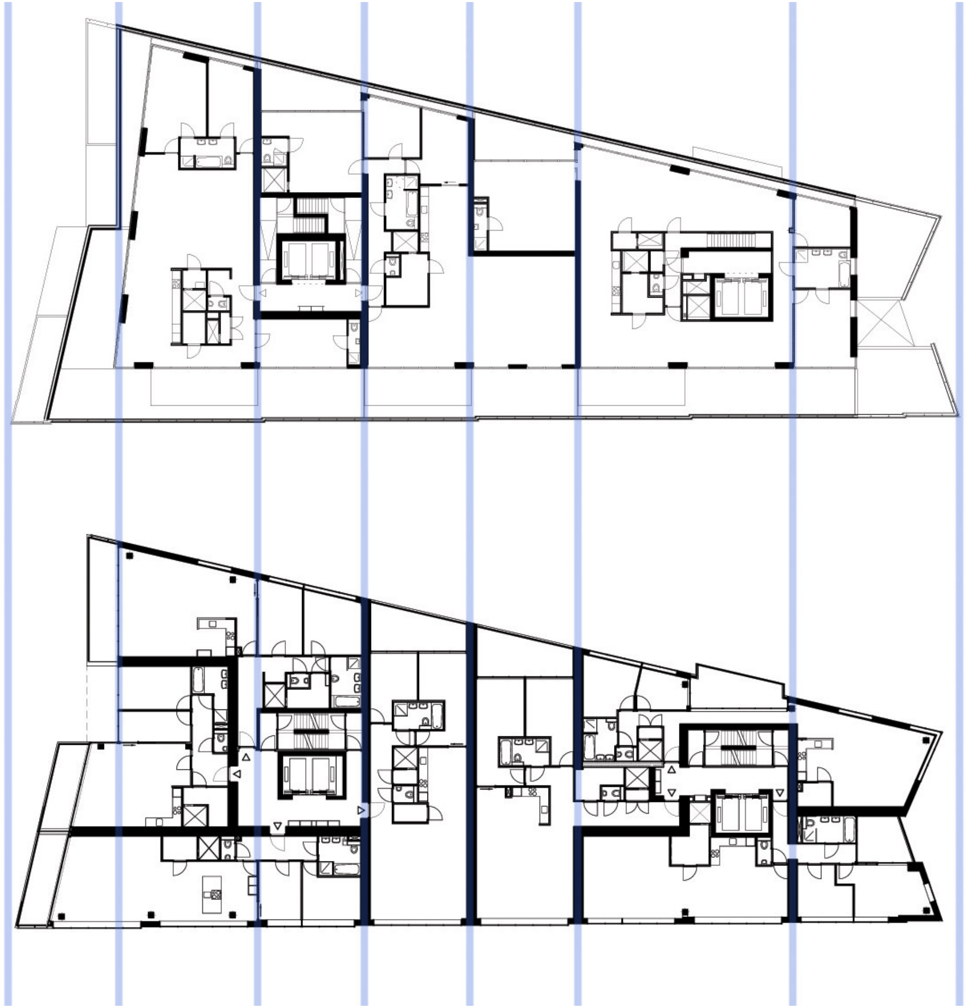


Fig. 3.1. Recognisable patterns in the 7th [below] and 23rd floor of the residential building La Fenêtre in The Hague [Rudy Uytenhaak Architectenbureau bv]

This research does not go into detail on the question whether grids are still the principal for designing a (multi-use) building, but with a topic on an optimisation process to connect different structural grids, this question might need an answer in the successor of this Master's thesis.

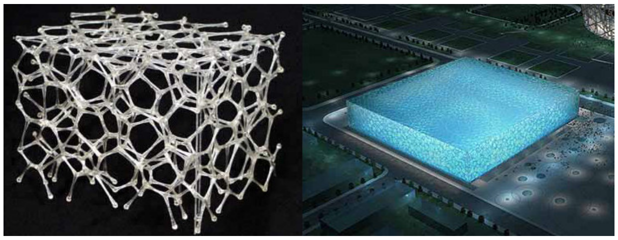


Fig. 3.2. The National Swimming Centre in Beijing [right] with a organic structure of soap bubbles for the exterior wall [www.archinect.com and www.arcspace.com, March 2006]

terminology

In the text below the terms, grid and modular dimension are used extensively. Therefore, four relating terms are explained shortly below [Figure 3.3, next page][9].

- (structural) grid
  a grid is a system of lines that serves as a tool for determining the location and direction of the structural elements. They also form an important aid in 'reading' construction drawings. Also; a network of evenly spaced horizontal and vertical lines that can be superimposed on a map, chart, etc., especially in order to locate specific points
- modular dimension
  modular dimensions are the distances between adjacent grid lines. It is possible to work with multiple modular dimensions in the same or in different directions
- centre line grid
  a centre line grid is a grid system consisting of singular lines. The distance between the (structural) elements placed on these lines are remaining dimensions
- band grid
  because of the fact that many elements are used repetitive in a structure, it can be obvious to make use of a grid that is tuned to these elements. A band grid is such a grid where zones are demarcated for the specific sizes of elements, where the centre grid lines may become obsolete.

---

[9] Scheers [18]

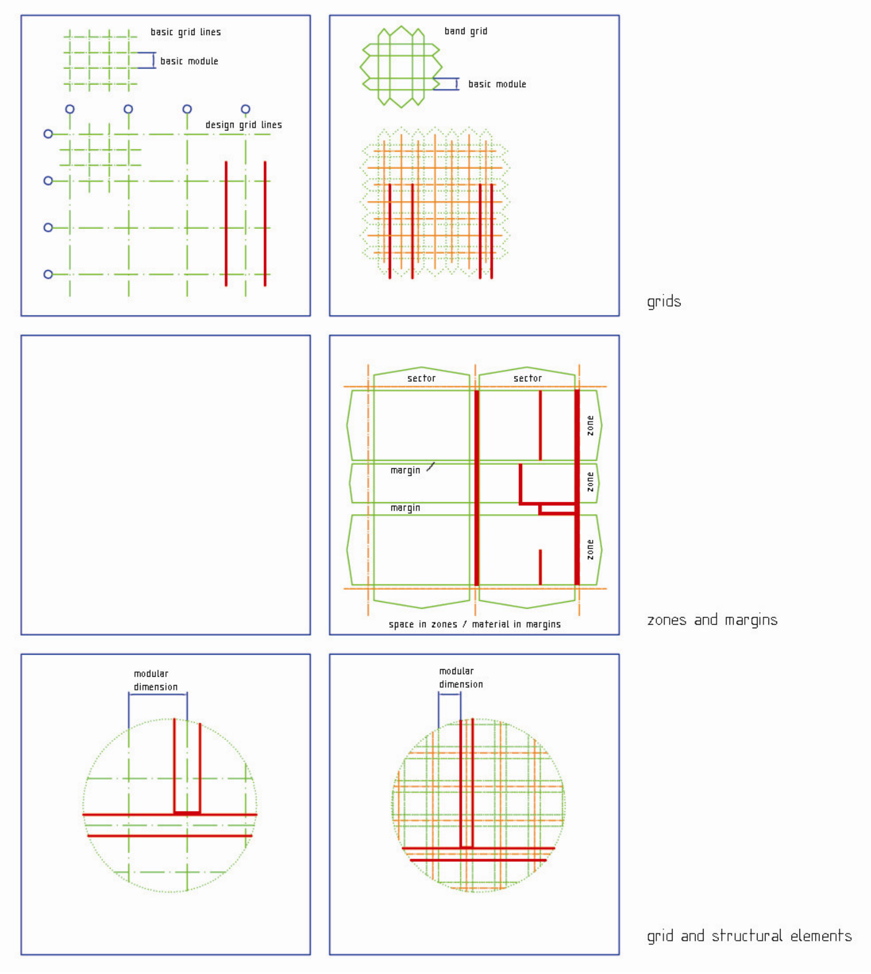


Fig. 3.3. The basic terms in design grid lines and modular dimension bands

3.2 modular dimensions of buildings

As stated in the previous section, modular dimensions are the distances between adjacent grid lines. Modular dimensions should be able to count as the largest common dimension of all other usable dimensions in a structure. This used to be a dimension relating with the size of a body part, e.g. the thumb, the palm of a hand, or the length of an arm. So not only absolute dimensions, but also relative dimensions were/are important in the building industry. Besides these two systems of dimensions, perceptual dimension also play an important role [Section 3.2.1]. Modular dimensions are not only used to measure, but they function also as a design tool. These design 'modules' are formed by multiples of the measuring module. Some simple examples are given below [Figure 3.4].

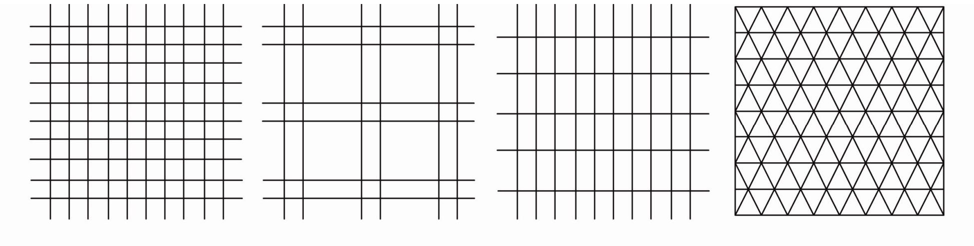


Fig. 3.4. Four simple examples of design modules

3.2.1 perceptual dimensions of structures

Despite the standardised measures (e.g. the metric system), people use their own dimensions as a perception for space and objects. This type belongs to the system of *perceptual dimensions*, this next to *absolute* and *relative* (e.g. the Modulor by Le Corbusier) dimensions and deals with dimensions that are dependent on or under the influence of factors like time, movement, location and circumstances. Examples of this type are perceptions that a standing object seems much larger than a laying object.

The storm flood barrier the 'Maeslantkering' near Rotterdam is twice as long as the highest building in Holland, the 'Delftse Poort' building in Rotterdam, although it might not seem that way [Figure 3.5]. And objects seem larger at night than in the daytime[10].

Fig. 3.5. The 'Maeslantkering' – approx. 300 metre in length [www.bus-idee.nl, December 2005] and the Delftse Poort – approx. 150 metre in height [www.skyscrapercity.info, December 2005]

---

[10] Scheers [18]

The dimensions considering structural grids belong to the system of absolute dimensions. The organisation of structures or structural elements can be neutral or directed. The latter type is characterised by the usage of one or more grids, whether or not in an orthogonal way, where one of the directions is dominant. Considering this, the designer of a structure could prefer a neutral organisation in order to create a feeling of *randomness*. An additional advantage of the neutral organisation is that it can easily be extended in all directions, especially when an orthogonal grid is used. As a basis for a structural grid in housing, normally a design modular dimension of 30 centimetres is used. However, for smaller spaces in a house (e.g. the toilet) or for larger spaces in utility building engineering this dimension will vary.

3.2.2 controlling dimensions

The controlling dimensions are dimensions between key reference planes. They provide not only a framework for design but also a basis which components and assemblies may refer to [Figure 3.6]. Standard dimensions are theoretical but, in practise, they provide the basis for individual, basic structural and finished measurements, linking all building components in an organised way.

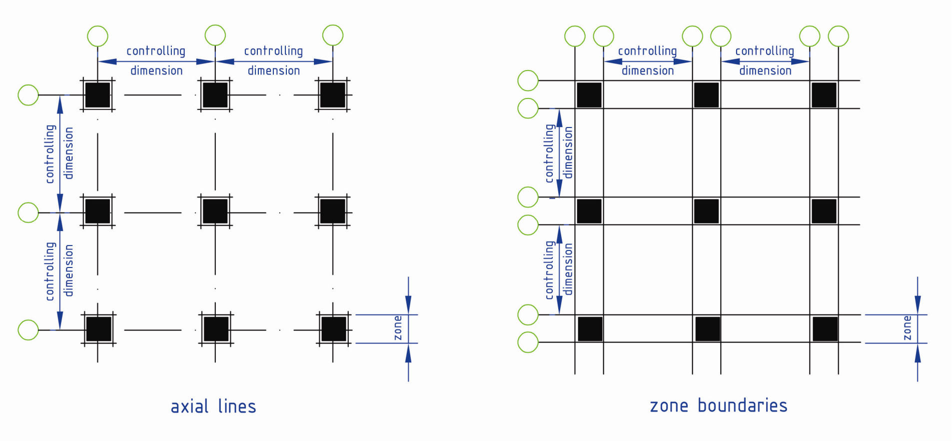


Fig. 3.6. Horizontal controlling dimension

3.2.3 modular dimensions of different building types

In this section the standard modular dimensions for three different building types are given. The modular dimensions of (underground) parkings are given in Section 2.2.2.

residential buildings

Many utilised modular dimensions for residential buildings (along the façade) are 4.80 m, 5.40 m, 6.00 m, 7.20 m, 7.50 m, and 7.80 m. But for apartments, generally larger dimensions are applied. On the grid lines, usually load bearing walls are placed, resulting in a wall skeleton. The general depth of the modular dimensions is between 8.00 m and 12.00 m and this dimension is much more flexible than modular dimensions along the façade.

office buildings

Regarding the modular dimensions of office buildings, generally a multiple of 0.60 m is used. This has its origin in the finishing aspects in an office, like ceiling elements, shafts, etc. A very common dimension along the façade is 7.20 m. The depth of an office often varies between 12.00 and 14.00 m. Normative for the dimensioning of the maximum building depth, the view is governing as stated in the Dutch Working Conditions Act, the 'Arbowet'. For hallways usually a width of 3.60 m or 4.80 metre is chosen. The structure of an office building can consist of a wall skeleton, a columns skeleton or an intermediate solution.

commercial buildings

For commercial building a distinction between two types can be made, boutiques and supermarkets or shopping malls. A standard dimension for a boutique is 6.00 m by 15.00 m with the smallest dimension along the façade. 5.40 m and 7.80 m are used as well for this dimension. For supermarkets and shopping malls, specific demands are given by the tenant/owner concerning the surface and modular dimension. In general, the modular dimensions should be as large as possible. This resulting from the wish to have a functional layout that is as flexible as can be. The most logical structural system for this kind of buildings is therefore the column skeleton.

3.3 structural grid layouts in multi-use buildings

3.3.1 geometric considerations

When considering geometry in a building, Neufert[11] states that by means of the system of coordinates, buildings and components are arranged and their exact positions and sizes are specified. The nominal dimensions of components as well as the dimensions of joints and interconnections can thereby be derived. A coordinate system consists of planes usually at right angles to each other, spaced according to the coordinate measurements. Depending on the system, the planes can be

---

[11] Neufert [14]

different in size and in all three dimensions. As a rule, components are arranged in one dimension between parallel coordinate planes so that they fill up the coordinate dimension, including the allowance allocated to the joints and also taking the tolerances into account. Hence a component can be specified in one dimension in terms of its size and position. This is referred to as boundary reference [Figure 3.7].

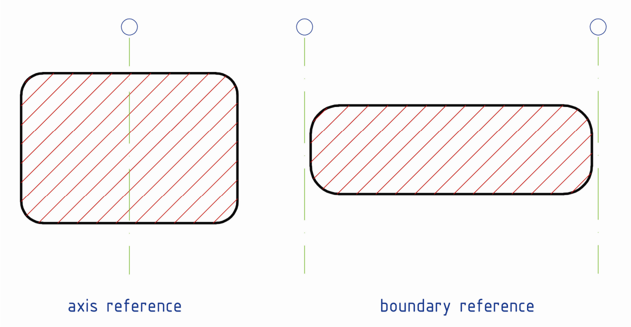


Fig. 3.7. Axis reference and boundary reference for grid layouts

In other cases, it can be advantageous not to arrange a component between two planes, but rather to make the central axis coincide with one plane of the coordinate system. The component is initially specified in one dimension with reference to its axis, but in terms of position only. A coordinate system can be divided into sub-systems for different component groups, e.g. load-bearing structure, component demarcating space, etc. It has been established that individual components need not be modularised, e.g. individual steps on stairways, escalators, windows, doors, etc. For non-modular components which run along or across the whole building, a so called 'non-modular' zone can be introduced, which divides the coordinate system into two-sub systems. The assumption is that the dimension of the component in the non-modular zone is already known at the time of setting out the coordinate system, since the non-modular zone can only have completely specified dimensions. Further possible arrangements of non-modular components are the so-called centre points and edge positions within modular zones.

3.3.2 common grids

Many different types of structural patterns are possible. In most buildings, a repetitive geometrical pattern or grid is usually present both in the vertical support system, whether it is formed of load-bearing walls, columns, or some combination of the two, and in the horizontal spanning system. The geometries and dimensions of individual units and the way these units aggregate in such buildings are invariably dependent on the programmatic requirements of the buildings. These same geometries and dimensions, in turn, strongly influence the types of structural systems that are most appropriate for use. It is useful first to review briefly some basic characteristics of patterns themselves. Obviously, many building are actually *single-cell* structures, e.g. gymnasiums, factory buildings and exhibition halls. Most buildings, however, are composed of a large number of *aggregated bays* or *units*. Several common types of structural patterns are evident in housing or office buildings. In a housing scheme in which a serial aggregation pattern is used tot arrange the basic units, the structural patterns of the vertical support system itself consists primarily of a series of parallel lines. These lines in turn, could consist of a series of load-bearing walls or a column-and-beam system arranged in the same pattern. A common square grid or a *two-way aggregation* can be used as well. A known example of how the structural pattern of a building can be related to, and reinforce how a building is functionally zoned, is the *tartan grid* of discrete vertical support elements [Figure 3.8][12].

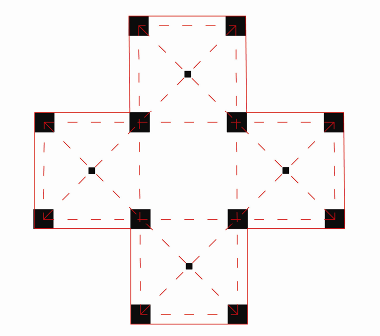


Fig. 3.8. An example of a tartan grid

Square or rectangular grids are common but other geometries are frequently encountered as well. Systems that are based on any elementary geometric patterns may be used, especially when the overall shape of a building calls for an obvious non-orthogonal response. It should also be noted that common shapes may be framed in significantly different ways.

Grids are often altered in zones where something special occurs, such as a large column-free space within a building, or at the boundaries of the building. In situations where the pattern of the vertical support system is irregular, some structural systems are more attractive than others. Totally irregular grids result in a decrease of effective or economic use of systems involving a number of highly repetitive and typically off-site-produced, modular units, e.g. precast concrete elements. For full advantage to be gained from the construction process possibilities inherent in such systems, they must be used in an almost brutally repetitive way. Poured concrete systems are usually preferable in cases involving irregular patterns, since irregularities can be more easily handled.

---

[12] Schodek [20]

3.3.3 meeting of structural grids[13]

Pattern intersection can be a problem with respect to using one grid on top of another. This situation quite often arises when it is desired to place a roof system having a grid geometry on top of an internal system having a different geometry. In such cases, a strategy of alignment, together with vertical supports common to both systems, is used [Figure 3.9, left]. Occasionally, a bypassing type of approach is used wherein the vertical supports for the upper grid are maintained differently from those of the lower grid [Figure 3.9, right]. The systems are thus independent of one another. The former alignment strategy is commonly used when the overlapping grid patterns are fairly similar to one another. The latter bypassing strategy is frequently used when the superstructure is a relatively long span compared with the substructure, of simple overall geometry, and has an internal pattern different from that of the substructure. The bypassing structures need not be of the same material, although they can be. When the overlapping patterns are similar, usually the same materials are often used.

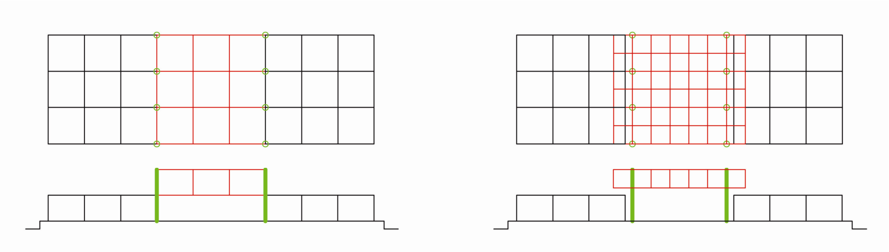


Fig. 3.9. Basic strategies for the layering of structural patterns. Alignment of grids [left] and the bypassing of grids

In many cases, more than one generalised structural pattern is used in a building. The reasons for this multiplicity of patterns vary. Quite often, there is a wide variation in the programmatic requirements of the building such that a variety of minimum clear span dimensions exists. One or more standard functional patterns may be adopted to respond to this variation, instead of a single pattern that compromises the varying requirements present. In other cases, physical constraints, such as non-uniform foundation conditions, may dictate using different structural grids in different areas. When more than one generalised structural pattern is used in a building, the way in which the patterns meet becomes a basic structural design issue. Intersection points always call for unique treatment or special elements. Adopting multiple patterns, however, requires finding ways of rationally joining dissimilar systems. Consider the two structural grid systems shown in Figure 3.10. It is evident that if these two systems randomly intersected, the structure in the region of the intersection could not be characterised as typical of either general grid. If a beam was used along the boundary, for example, the beam would

---

[13] Schodek [20]

be loaded in a way that is different from those present in either the grid to the left or the one to the right. It would therefore have to be designed differently. Additional columns might also be needed. While randomly intersecting grids are certainly possible, other strategies may prove preferable.

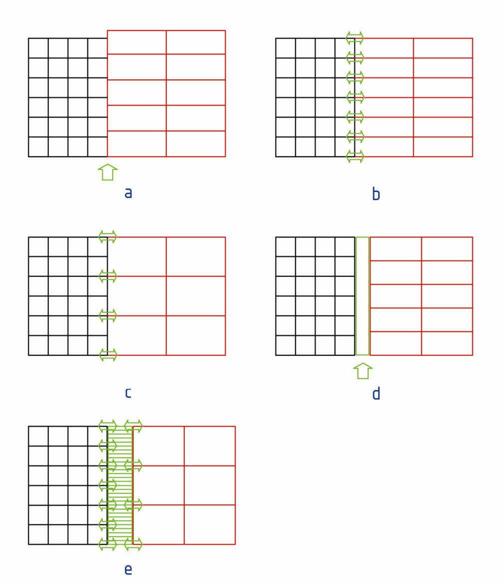


Fig. 3.10. Basic strategies for the meeting of generalised structural patterns

Figure 3.10 illustrates several basic approaches to pattern intersections. Other approaches exist as well. One basic approach is illustrated in Figure 3.10b and 3.10c would be to use a strategy of alignment in which the smaller grid is designed either as directly related to, or as a subdivision of the larger one. This alignment reduces the misfit between the systems and does not require additional vertical supports.

Another general approach is to use some sort of mediating system between the two general systems. One type of mediator would simply be a strip of space itself, in which case the whole problem is simply bypassed, see Figure 3.10d. Another type of mediator would be a third structural system of some sort. This third system must necessarily reflect or be adaptable to the characteristics that are common to both of the general systems, or else it must be extremely flexible in some other way. A structural grid that is much smaller than either of the primary systems, but a logical subdivision of each is often used to join the primary systems, see Figure 3.10e. Smaller grained systems of this sort are often, but need not to be, made of materials different from that of the primary systems. The primary systems, for example, might be constructed of poured reinforced concrete, while the smaller grained mediating system might be a light-steel system. Mixing of materials in this controlled way is usually quite acceptable, even in purely economic terms.

Intermixing materials within primary systems, however, is usually more difficult. An advantage of using a small-grained mediating system is that the designer has greater freedom in placing the primary grids. The primary grid lines do not have to be aligned. Load bearing walls are also often used as third element mediators, particularly when the primary systems are not aligned. However, a load bearing wall can be taken as a continuous grid having infinitesimal grid spacing and worked with accordingly.

A structural grid based on zones for elements (i.e. the band grid) could be unfavourable if multiple spaces have to be connected, especially when those spaces are on top of each other [Figure 3.11][14].

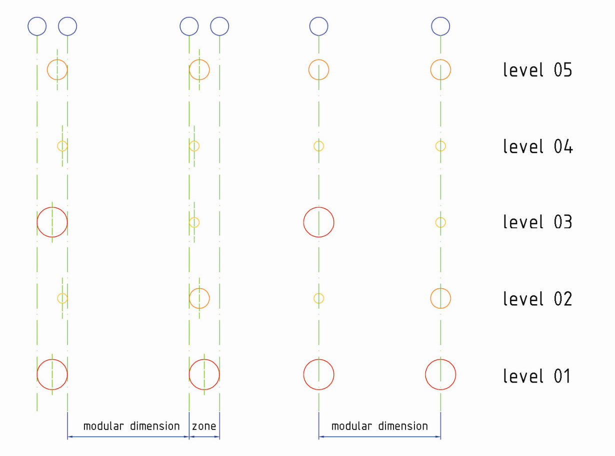


Fig. 3.11. Unfavourable layout of the structural elements with a band grid

3.4 concluding remarks

As mentioned in the introduction of this chapter, the process of designing a structure is implicitly linked to the process of designing the overall building, including the functional layout, and thus the design of the grid system. This makes combining different dimensions a main aspect of structural design, with both perceptual (in this Master's thesis intensified in creating a feeling of randomness) and absolute dimensions. Based on different references, the grid systems can be found on various predefined layouts, or can be arranged for one specific space as well. When a number of these aspects need to be combined in one building, the meeting of various structural grids plays an important role in the overall performance of that building.

---

[14] Scheers [18]

4 designing the structural optimisation tool

This chapter deals with the designing of the optimisation tool for the allocation of load bearing columns in transfer zones of multi-use buildings. Section 4.1 and 4.2 deals with the several characteristics the problem has and the tool has to posses, whereas Section 4.3 treats the specific aspects of this design tool.

4.1 characteristics of the problem

This section deals with the overview of the problem, including the layout of the problem [Section 4.1.1] and the reason of dealing with it, the determination of the optimisation aspects of the problem in Section 4.1.2 and other aspects that have to be dealt with [Section 4.1.3].

4.1.1 layout of the problem

As was stated in the introduction of this Master's thesis, the main question was in what way and how well a computational structural optimisation technique can be implemented in a design tool based on simple rules that will be used in the initiation of the design process for the allocation of load bearing structural elements of an intervening floor.

If the top and lower floors have the same or similar structural layouts [Figure 4.1 on the next page] the logical way of connecting these floors is by vertical elements strictly following this layout. In this Master's thesis, initially, two dissimilar structural floors will be connected with structural elements; e.g. the top layer of an underground parking and the lower layer of a residential or office building, similar to the three buildings in Figure 4.2., on the next page. The structural elements will be formed by columns of different size and with different angles. As was stated in the Master's thesis of E. Quispel [Section 2.2.2], this intervening structure can then replace an inefficient monolith transfer zone.

Fig. 4.1. Unité d'Habitation, Berlin by Le Corbusier [http://www.fotodeponie.de/, March 2006]

Fig. 4.2. From left to right: La Fenêtre, The Hague [Rudy Uytenhaak Architecten BV], Kadoorie Biological Sciences Building, Hong Kong [www.hku.hk, January 2006], No. 1 Deansgate, Manchester [www.ma.man.ac.uk, March 2006]

Figure 4.3 shows how the vertical load bearing elements of the superstructure are connected with the elements of the substructure via structural elements (cores and columns) for the 'La Fenêtre' building and the Kadoorie Biological Sciences Building.

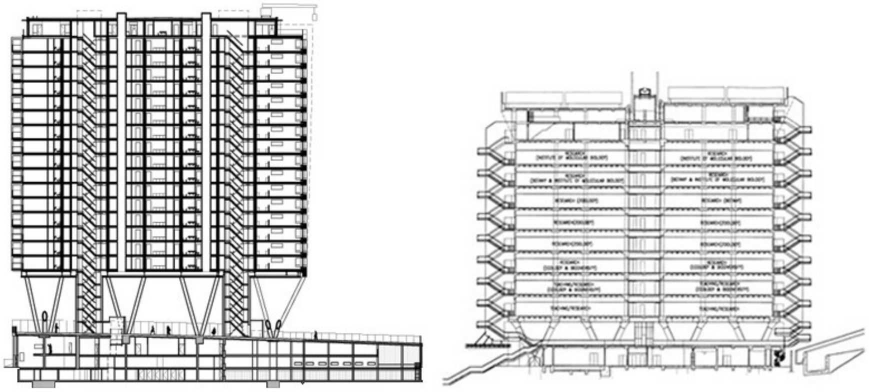


Fig. 4.3. Vertical sections of La Fenêtre, The Hague [Rudy Uytenhaak Architecten BV] [left] and Kadoorie Biological Sciences Building, Hong Kong [www.building.com.hk, March 2006]

For the given buildings, the columns follow a rather strict pattern. The goal of the usage of an optimisation tool is to create a sort of randomness [Section 4.1.3] in the intervening floor, similar to the column onset of 'The Componist' by ONX Architects [Section 2.2.1].

4.1.2 the optimisation aspect of the problem

Every design has a certain measure of success, or fitness considering the purposes it needs to fulfil. When dealing with optimisation this fitness has to be denoted explicitly. In many cases, the bearing capacity of a structure versus the size of the structural elements is used to perform an optimisation routine.

With the problem of allocation structural elements between two structural grids with their accompanying functional layout, a mere optimisation on load versus size would not be satisfactorily. The structural layout of the intervening floor must not be of negative influence of the functional aspects of the floor above and under it. In fact, it should make sure that the functional arrangement of the grid of these floors is as optimal as possible. In other words, the optimisation aspect of this problem should be concerned with the functional arrangements of both the intervening floor and the top and lower layer of the sub- and superstructure it connects.

4.1.3 other aspects concerning the problem

"To prevent architectural solutions for structures from being interpreted by the public as boring, repetitive or even ugly, not solely architectural treatment needs to be persuaded. A structural design approach could also lead to new and improved shapes and appearances of structures." Prof. Dr. Jiří Strasky, principal bridge designer, claims that the way to improve the architecture of bridges is to produce better structural solutions. "In bridge design the responsibility for producing better architectural solutions lies not with architects but with structural designers. [...] Some criteria of good structural solutions are also criteria for good architectural solutions. A good architectural solution must be *a priori* a good structural solution" and "in designing a bridge, a structural form should be considered appropriate as a structural solution only when the design uses the inherent structural characteristics of the form to its advantage[15]."

These statements can be well implemented in the solution to the problem that is dealt with in this Master's thesis. As an initiation of the design process, the computational design tool can easily give solutions that are not foreseen, instead of solutions that are logical or could be found mathematically, creating new insights, and possible new architecture. Moreover, the interesting thing about this is that it is not only applicable on (expensive) pavilions that have to serve their function for a limited period of time, but on almost any multi-use utility building.

randomness

The aspect of unforeseen solutions can also be adopted when talking about the randomness of the allocation of the columns. The expectation is that the design tool will create an intervening structure between two functionally different floors with a random allocation to the eye, similar to the new National Stadium of Beijing [Figure 4.4, next page]. This design was created with the random appearance of the steel structural elements as main characteristic. However, the main joists of the structure are following a rather strict layout, but due to the random location of the connecting beams, the total façade, and roof the total structure looks like one big birds nest. Actually and surprising as it may seem, it is difficult to get a computer (or a person) to do something by chance, or in other words to 'create randomness'. A computer running a program follows its instructions blindly and is therefore completely predictable. This means for the load bearing elements for this Master's thesis that they, similar to the birds nest structure of the National Stadium of Beijing, only look randomly placed, but actually are placed based on predefined algorithms and to pre-appointed possible locations.

One could say, randomness is a lack of structure. In this Master's thesis, this paradigm makes the wish for randomness in a building quite problematic. Therefore, the term randomness is replaced by *variation*, meaning that there needs to be a variation between the different boundary conditions for the column parameters. For a detailed explanation on the variation of the column parameters, see Section 4.6.2.

---

[15] Strasky [21]

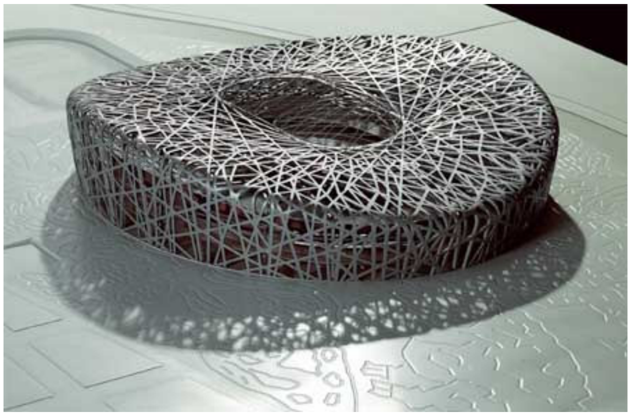


Fig. 4.4. The National Stadium of Beijing for the 2008 Olympic Games by Herzog and De Meuron and Arup [www.chinadaily.com.cn, March 2005]

multi-storey functional connections

Another aspect of the allocation of columns based on an optimisation algorithm is the possibility to expand the application of it over more storeys, instead of just using it for a single transfer zone, and thus making it useful for connecting different functional layout within a multi-use, multi-storey building. A structural appearance similar to the Sendai Mediatheque [Figure 4.5] would then be possible. In this Master's thesis, this option is not examined.

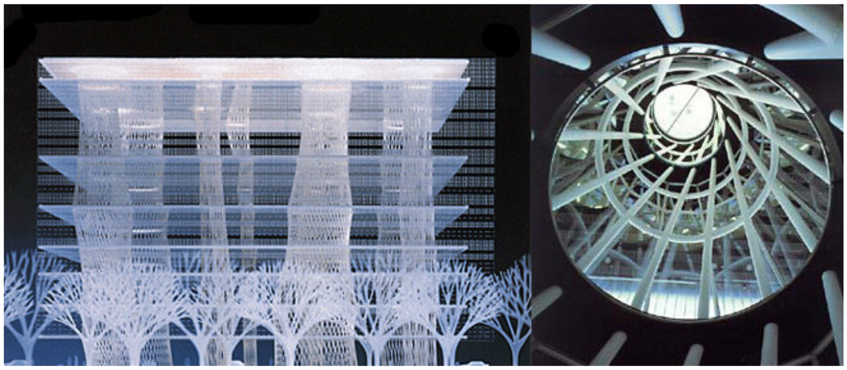


Fig. 4.5. A model of the Sendai Mediatheque [http://archive-www.smt.city.sendai.jp, December 2005] and a photo of the internal open column [www.designboom.com, December 2005]

4.2 characteristics of the design tool

optimisation in the design tool

It could be one's opinion that in general optimisation problems are unsolvable. Very often a choice has to be made between a 'good' model, which cannot be solved (although it might be tried), and a 'bad' model, which can be solved for sure. For instance, the most widespread optimisation models now are still *linear optimisation* models[16]. But it is very unlikely that such models can describe the nonlinear real world very well. Indeed, the real life is too complicated to believe in a universal tool, which can solve all problems at once.

Performance of a method could be described with an intuitive definition like the total amount of *computational effort* that is required by the method to solve the problem. But what is meant with '*solve the problem*'? This could mean finding an *exact* solution, however in optimisation, this seems impossible. Therefore, it could be stated that to solve a problem means to find an *approximate* solution with an accuracy of $\epsilon > 0$. This, as a result of the introduction of a stopping criterion, which is used to reduce the required computational effort, or in other words the time needed to come to a solution.

It should also be noted that optimisation includes the compromising of different solutions based on the constraints as is explained in Section 2.1.1 of book two of this Master's thesis. Not all problems can be solved within one design tool or better said, the perfect solution to all difficulties in designing a structure is never to be found. Besides, if too many problems have to be solved, the actual problem will be lost out of sight. That is why the tool to be designed for this Master's thesis will focus on a limited number of problems, trying to solve them as good as possible, and thus creating a start point for the actual design of a transfer zone between two or multiple structural grids.

modelling assumptions

As in every model, there needs to be considered how well the real world will be described. And in every model, assumptions needs to be made, as a model is a simplification of the real world by definition.

---

[16] Engelbrecht [5]

Below some modelling assumptions are given, that can be helpful in reducing the complexity of the problem.

- the columns are pinned at the top and the bottom, resulting in the fact that the columns can be loaded only axially;
- every column is homogeneous, with an axial stiffness depending on the cross section area and the properties of the chosen material;
- the horizontal load from the individual slanting columns should be taken up by the floors;
- only columns are used as load bearing elements;
- the sub- and superstructure are rigid bodies.

4.3 aspects of programming the design tool

4.3.1 general aspects of the design tool

In book two of this Master's thesis numerous possible operators and aspects for the genetic algorithm have been given. One of the questions that remain to be addressed is how to select the operators and aspects to be used, as well as how to deal with their corresponding properties. Besides, as the optimal combination of the influences of the fitness parameters is problem dependent, extensive simulations have to be conducted to find the optimal weighing combinations[17], which is a time consuming process. The goal could be to try to get as much constraints as possible already in the solutions, in order to limit the time needed to check the solutions for their fitness. In the case of the GA script on the allocation problem of columns, the maximal column length and the restriction that the centre lines of the columns cannot coincide are implemented as constraints before the actual evolutionary process. Next to this, because design always mediates between many goals and hardly ever fits one of them perfectly, users could be given the ability to diverge from the optimal structural solution to achieve greater design flexibility, which is possible due to the output in MS Excel worksheet where the values of the several fitness parameters are given.

To start off with the process of designing an optimisation tool, simple rules should be used in the program, as was stated in the problem formulation, and the problem should be simplified as much as possible.

---

[17] Engelbrecht [5]

If the outcome of the 'simple' optimisation procedure is satisfactory other aspects like

- non-rectangular grids
- curved grids
- multiple storeys
- multiple load cases
- structural dynamics
- fire loading
- etc.

can be implemented in the design tool.

4.3.2 structural phenomena and general design responses

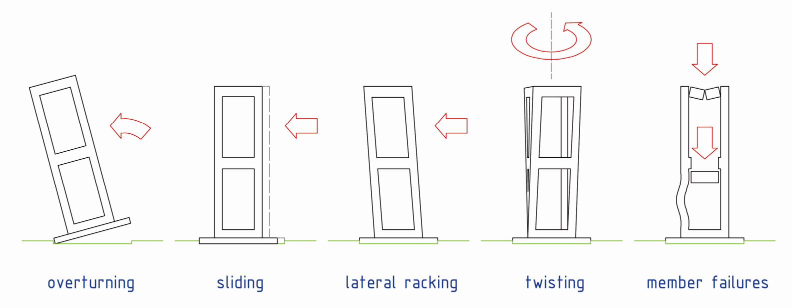


Fig. 4.6. Structural phenomena of a 3D-framework

All structural failing phenomena [Fig. 2.1] – overturning, sliding, raking, twisting, bending, shear, and buckling – occur as a direct or indirect consequence of a force acting on a whole structure or some specific component within it[18]. It is obviously necessary to understand quantitatively or numerically the type and magnitude of these forces in order to determine whether a structure is susceptible to failure in any of the modes noted previously, or, alternatively, to determine how large a member should be to carry expected forces safely, or in other words a process called structural analysis and design.

---

[18] Schodek [20]

For the GA script, two important steps in the structural analysis need to be made.

step 1.
determination of the centre of gravity as point of application of the forces
step 2.
determination of magnitudes of the forces acting on the whole structure

When expanding the model with an analysis of the possible structural phenomena, a third step needs to be made.

step 3.
determination of how the analysed forces might cause the structure to fail according the phenomena mentioned above, and how the external forces cause internal forces to develop within parts of the structure

Concerning the second step, it will be possible for the designer to give the values of the forces acting on the structure in the AutoCAD graphical user interface (GUI). This includes the vertical point loads acting as the summed vertical dead surface weight and the variable surface load, the horizontal point load in the direction of the grid lines acting as a resultant of the horizontal wind, the horizontal point load perpendicular to the grid lines also from the wind load, and finally the moments in three directions as resulting forces from the translated vertical and horizontal loads [Figure 4.7].

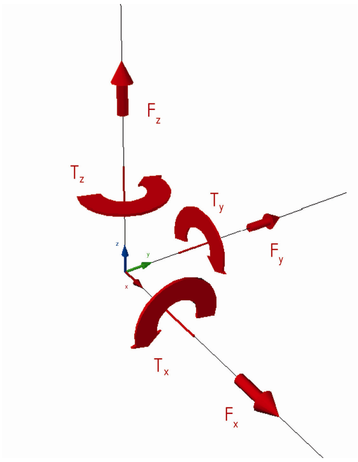


Fig. 4.7. Input of loads on the structure

The design tool itself then has to check whether the column structure can cope with the external and internal forces, or in other words, the tool is responsible for the actions that have to take place in step 2.

4.3.3 analysis and design criteria beyond the scope of the design tool

Not all possible criteria can be implemented in the fitness function. The main problem is how to define every aspect used in designing a structure. First of all, in the design phase not all input values for all criteria are known. Remember that the goal of the Master's thesis is to create a design tool that can be implemented in the conceptual and preliminary design of a structure. Secondly, in many cases it is impossible to define subjective aspects of a structure, e.g. social safety and aesthetics. Besides, with an increasing number of fitness parameters, the difficulty of finding a solution that has acceptable values for all parameters becomes larger and larger.

Below, some criteria are given to which the structure has to reply to, next to the ones in the fitness function, but which won't be utilised.[19]

serviceability

The structure must be able to carry the design load safely, and without excessive material distress and with deformations within an acceptable range. This also includes movements in structures, or accelerations and velocities of structures carrying dynamic loads, in other words, the stiffness (including the damping) of a structure.

construction

Construction criteria are diverse and include such consideration as the amount and type of effort or human power required to construct a given facility, the type and extent of equipment required, and the total amount of time necessary to complete construction. The complexity of a structure is usually read from the number of the different structural elements and the relative degree of effort necessary to assemble the pieces into a whole, which is influenced by the size, shape and weight of the elements. Of course, also repetition of the construction operations plays an important role.

costs

The cost criterion cannot be separated from the criteria of efficiency and construction. The total cost of a structure depends primarily on the amount and cost of material used, the amount and cost of labour required to construct the structure, and the cost of equipment needed during construction.

Of course, some of these aspects are implemented in the evaluation of the solutions, but in a less recognisable manner. For instance, the total volume of the material being used is somewhat related to the cost of the structure. These three

---

[19] Schodek [20]

criteria focus on the analysis and design of structures in a structural context, and can be determined objectively. However and as already mentioned, the total value of a building is also determined by other (more subjective) aspects of which aesthetics and functionality are the most important.

4.3.4 analysis and design process for structures

According to Schodek[20] the following seven actions need to be undertaken in order to design a structure.

1. *geometry definition*. The basic geometry of the structure needs to be drawn, with particular attention paid to member hierarchies.
2. *load assessment*. The load assessment involves determining loadings associated with the dead loads and live loads on the structure resulting from its self-weight of the structural members and the external loads as loadings due to wind.
3. *modelling of the structure and boundary conditions*. The modelling of the structure includes characterising complex real-world construction connections as one or another of an idealised set of supports.
4. *load modelling*. Distributed surface loads acting over an area are converted into loads that act along the length of a member. Other loads are idealised as concentrated or point loads.
5. *determination of reactive forces*. With the application of equilibrium principles, the set of reactive forces and moments that are developed at the boundaries of the structure as a consequence of the external loading are developed. In order to do this, first the structure needs to be classified as a statically determinate structure – no more than three unknown reactions are involved, the reactions can be found through simple application of the basic equations ($\Sigma F_x = 0$, $\Sigma F_y = 0$, and $\Sigma M = 0$) – or as a statically undetermined structure.
6. *determination of internal forces and moments*. The internal forces and moments are developed in the structure as a consequence of the action of the external forces.
7. *determination of adequacy of the structural members*. If the internal forces and moments states are known, it can be determined whether the actual member used is adequate to carry the forces and moments involved.

These steps more or less describe the process that is common to analysing any type of structure. But all steps require judgement. Each element in the building must be reviewed with respect to its potential contribution as a constituent part of the whole structure. But all these steps need to be found in the script of the design of the column structure.

---

[20] Schodek [20]

4.3.5 structural and functional aspects of columns

failing of columns

Columns, in their most common application, are vertical support elements. But strictly speaking, columns need not to be only vertical. They are rigid linear elements that can be inclined in any direction, and to which loads are applied in general at member ends[21]. Columns are normally not subject to bending. Columns tend to fail by crushing, dependant of the yield stress ($f_{y;d}$) and the surface area of the column (A) or by buckling, dependant of the moment of inertia (I), the Young's modulus of the column material (E) and the length of the column (l_{buc}). It is assumed that buckling can only be governing with steel section, not with concrete sections. The Euler buckling load is described as

$$F_E = \frac{\pi^2 EI}{{l_{buc}}^2} \tag{4.1}$$

Also, the nature of the end conditions of the column is of importance. If the ends of a column are free to rotate, the member is capable of carrying far less load than if the ends are restrained. Also bracing the element in some way increases the stiffness. Another important failure mode for columns is overturning. An externally acting horizontal force on columns tends to cause the columns to overturn. For a column not to overturn there must be some counterbalancing rotational moment, which in the case of vertical (or almost vertical) columns is partially provided by the dead weight of the column itself [Figure 4.8]. If the rotational moment as a result of the dead weight of the column works in the same direction as the moment as a result of the externally horizontal force, then there can be equilibrium between the two moments and the column will overturn.

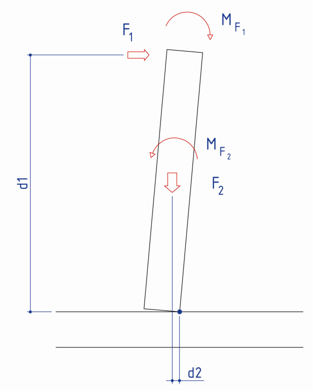


Fig. 4.8. Mechanism of overturning of columns

---

[21] Schodek [20]

general principles of designing columns

For transferring the horizontal and vertical load, the number of columns and the column properties – the surface area A, the length l, and/or the angle θ – need to have adequate values. A practical preliminary design for a structure capable of transferring these loads is based on logical and efficient load transfer and an economical use of materials, which is determined by the measurements of the different structural elements[22]. In other words, a vicious circle occurs. In order to escape from this circle, it is needed to make use of certain rules of thumb, with which the dimensions can be determined sufficiently accurate. Besides, detailed calculations in the preliminary design can slow down the creative process.

For the purpose of the design tool, it can be assumed that the most simple column type can be applied:

- the buckling length (l_{buc}) of the column hinged on the top and bottom equals the system length (l) of the column;
- the column is loaded with an axial **compressive** force.

Therefore, besides the fact that columns need to be dimensioned on strength, the buckling 'problem', which is an aspect of the stability of the total structure, needs to be taken into account as well. [In the VBA-script, only buckling of steel columns can be taken into account]

A general column design objective is to support the design loading by using the smallest amount of material or, alternatively, by supporting the greatest amount of load with a given amount of material. Whenever the buckling phenomenon is governing, the full strength of the material used in the compressive member is not being exploited to the maximum degree possible. As a result of irregularities by physical and geometrical non-linear behaviour of steel columns, the actual buckling stress (σ_{buc}) is usually smaller than the Euler buckling load (σ_E). NEN 6770 gives four buckling curves – for different types of profiles – with which the buckling factor (ω_{buc}) can be determined [Figure 4.9, next page]. This buckling factor is dependant of the relative slenderness ratio (λ_{rel}) of the column, which can be calculated by taking the square root from the quotient of $N_{pl;d}$ and F_E.

$$\lambda_{rel} = \sqrt{\frac{N_{pl;d}}{F_E}} \qquad 4.2$$

---

[22] Raven [x]

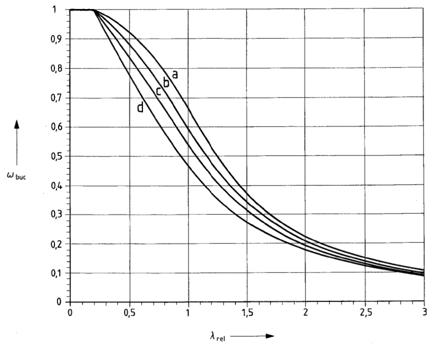


Fig. 4.9. Buckling curves for steel columns[23]

Table 4.1 shows the values of the buckling curves for different relative slenderness ratios.

	values of ω_{buc}			
λ_{rel}	a	b	c	d
0,2	1,00	1,00	1,00	1,00
0,3	0,98	0,96	0,95	0,92
0,4	0,95	0,93	0,90	0,85
0,5	0,92	0,88	0,84	0,78
0,6	0,89	0,84	0,79	0,71
0,7	0,85	0,78	0,72	0,64
0,8	0,80	0,72	0,66	0,58
0,9	0,73	0,66	0,60	0,52
1,0	0,67	0,60	0,54	0,47
1,1	0,60	0,54	0,48	0,42
1,2	0,53	0,48	0,43	0,38
1,3	0,47	0,43	0,39	0,34
1,4	0,42	0,38	0,35	0,31
1,5	0,37	0,34	0,31	0,28
1,6	0,33	0,30	0,28	0,25
1,7	0,30	0,28	0,26	0,23
1,8	0,27	0,25	0,23	0,21
1,9	0,24	0,23	0,21	0,19
2,0	0,22	0,21	0,20	0,18
2,1	0,20	0,19	0,18	0,16
2,2	0,19	0,18	0,17	0,15
2,3	0,17	0,16	0,15	0,14
2,4	0,16	0,15	0,14	0,13
2,5	0,15	0,14	0,13	0,12
2,6	0,14	0,13	0,12	0,11
2,7	0,13	0,12	0,12	0,11
2,8	0,12	0,11	0,11	0,10
2,9	0,11	0,11	0,11	0,09
3,0	0,10	0,10	0,10	0,09

Table. 4.1. Buckling factors

---

[23] Lutke Schipholt [4]

The values of Table 4.1 and the buckling curves of Figure 4.9 can be respectively calculated and drawn with the following formula, making it relatively easy to determine the values of ω_{buc} in the VBA-script.

$$\omega_{buc} = \frac{\sigma_{buc;d}}{f_{y;d}} = \frac{1+\alpha_k(\lambda_{rel}-\lambda_o)+\lambda_{rel}^2}{2\lambda_{rel}^2} - \frac{1}{2\lambda_{rel}^2}\sqrt{\{1+\alpha_k(\lambda_{rel}-\lambda_o)+\lambda_{rel}^2\}^2 - 4\lambda_{rel}^2} \qquad \ldots 4.3$$

The values for the factors α_k and λ_o can be derived from Table 4.2, with the buckling curve for hot-rolled circular hollow steel sections being curve a.

	buckling curve			
	a	b	c	d
α_k	0,21	0,34	0,49	0,76
λ_o	0,20	0,20	0,20	0,20

Table 4.2. Values for factors α_k and λ_o for formula 4.3

The text above shows that the primary factor of importance in connection with buckling is the slenderness ratio of a column; $\lambda = l_k/i$, with $i = \sqrt{I/A}$. The most efficient use of material can be achieved by either minimising the column length or maximising the value of the radius of gyration, for a given amount of material. Either or both will reduce the slenderness ratio of a member and hence increase its load-carrying capabilities for a given amount of material. Determining the required cross-sectional shape for a column intended to carry a given load is a task that is conceptually straightforward. The problem is similar to that of designing beams, wherein the objective is to find a section that provides the greatest moment of inertia for the smallest amount of material. In columns design, the task can more complicated because of the need to consider radii of gyration about different axes. This, of course, is of no importance when circular (hollow) sections are used.

4.3.6 structural mechanical aspects of complete system

kinematical/static (in)definiteness of structures[24]

The structure of the transfer zone needs to be constraint in such a manner that all free motions are prevented, or in other words the structure needs to be kinematical definite. If the support reactions and connection forces of a kinematical definite structure can be determined only with the use of equilibrium equations then the structure is statically definite. If a structure is statically indefinite, the deformations of the structure need to be involved in the calculation as well. So, whereas the kinematical definiteness needs to be guaranteed for every structure, the state of static definiteness is of concern considering the amount of calculation effort.

In designing the structural transfer zone, both the sub- and the superstructure are dimensionally stable, meaning that the structures preserves it original shape when it is disconnected from its supports. Next to this, if the occurring deformations of the sub- and superstructure are neglected, the structures can be interpreted as rigid bodies.

A rigid body in a plane has three degrees of freedom: two components of a translation and a rotation. Adding another dimension, the degrees of freedom consist of three translation components in the direction of the axes, and three rotation components around the three axes. In a plane, for every hinged column added to the rigid body, one degree of freedom can be dismissed. So with (at least) three hinged columns the rigid body can be kinematically definite in one plane [Figure 4.10, left].

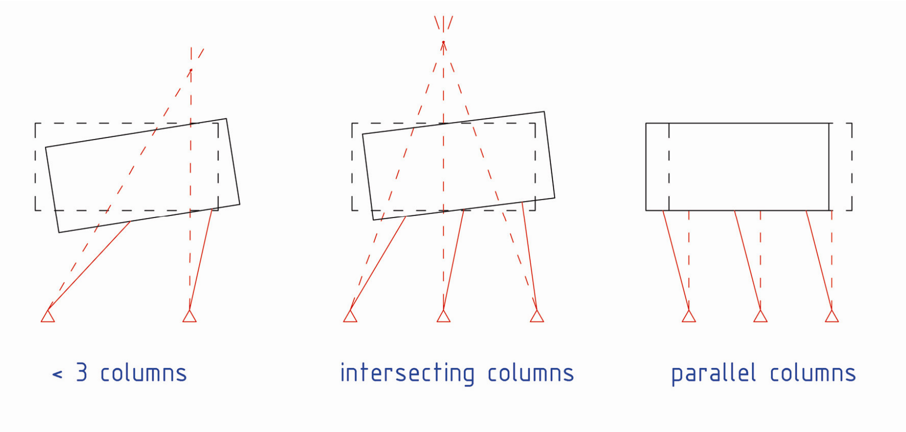


Fig. 4.10. Mechanism of overturning of columns

---

[24] Hartsuijker [8]

For a rigid body in a 3-dimensional space, at least six columns are needed to ensure that it is kinematically definite. Besides the number of columns, two other limiting conditions have to be satisfied. Both in the 2D plane as in the 3D space, the rigid body can rotate if the linear forces of the hinged columns intersect in the one point [Figure 4.10, middle], and can translate perpendicular to the direction of the hinged columns if all columns are parallel in relation to each other [Figure 4.10, right].

equilibrium of a rigid body and equilibrium of forces

Equilibrium demands that a rigid body does not translate nor rotate. In other words, to guarantee the equilibrium of a rigid body it must satisfy the force equilibrium and the moment equilibrium. In the model used in this Master's thesis, it is assumed that the sub- and superstructure are rigid and therefore are not liable to deformations themselves.

Equilibrium of forces of a rigid body in a 3-dimensional space demands that the resulting forces and the resulting moments in any point are zero.

$$\sum F_x = 0$$
$$\sum F_y = 0$$
$$\sum F_z = 0$$

$$\sum T_x \, \big| \, A = 0$$
$$\sum T_y \, \big| \, A = 0$$
$$\sum T_z \, \big| \, A = 0$$

The unknown support reactions have to answer to these six equilibrium equations. When six support reactions are at hand, then the reactions can be derived directly from the equilibrium equations. The support is then statically definite – there is an unambiguous solution to the problem of equilibrium.

If there are more than six support reactions, of which the work lines do not intersect in one point, and are not all parallel in relation to each other, then the number of unknown reactions is larger than the number of equilibrium equations, and there is an infinite number of solutions to the problem – the structure is statically indefinite. In this Master's thesis, the column forces are determined with the elongation of the columns based on the translation and rotation of the rigid superstructure. For the used model see Section 5.2.4, *calculation of the column normal force*.

Finite Element Method for bar structures

The Finite Element Method (FEM) for bar structures is internationally accepted as the most suitable approach for the calculation of strength and stiffness problems with the computer[25]. This method is also known as the displacement method, as the displacements have the main role as unknowns and as a result, they play an important role in the finite element method.

A characteristic of the method is that loads are only applied on the nodes of the structure and the support of the structure is rigid and cannot displace in vertical or horizontal directions. Rotations are not restrained.

For the design tool in this Master’s thesis, a method belonging to the standard 2D Finite Element Method-family for bar structures is used, called the *Discrete Element Method* (DEM). For more information on this method, see Section 5.2.4.

4.3.7 alternative FEM model for calculating the distribution of column loads

Another FE method that can be used for calculating the distribution of the column loads, is a method described in *Designing and understanding precast concrete structures in buildings* [22].

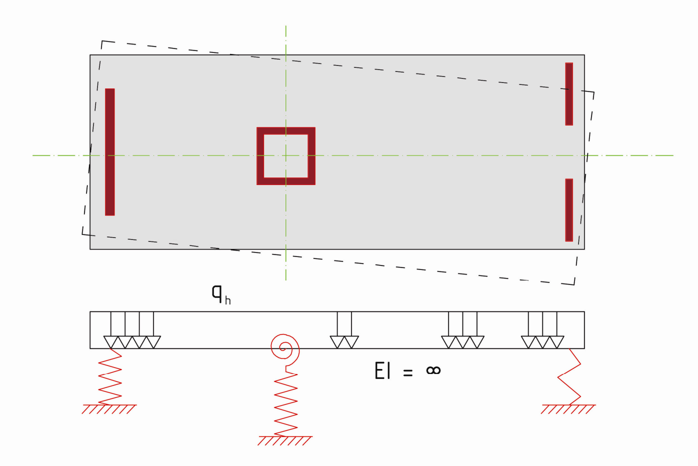


Fig. 4.11. A schematised representation of a infinitely stiff floor and walls

If the stability of a building is provided by several elements, the distribution of the lateral loads is based on the assumption that the floors are infinitely stiff in their own plain. Walls, being the stabilising elements, are schematised as springs [Figure 4.11]. The same principle can be used with vertical loads on beams supported by pin-connected columns.

---

[25] Hartsuijker [9]

In general, the building will translate and rotate. The distribution of the vertical load between the stabilising elements is depending on the **position** and **stiffness** of the bracing elements in the structure.
The relative stiffness of each bracing element determines the force carried by that element, and in turn defines the horizontal deflections in the floor plate.
Stabilising elements inclined to the direction of the load may be resolved into Cartesian components. The reaction in each stabilising element is given by the general expression[26]

$$H_n = \left[\frac{I_n}{\sum I_i} + \frac{e I_n a_n}{\sum I_i a^2} \right] H \qquad 4.4$$

where H = total applied load
H_n = reaction force in column n
I_n = moment of inertia of column n
I_i = moment of inertia of all columns
e = distance from the shear centre of all columns to the work line of the externally applied load
a = distance from the shear centre of all columns to the work line of each column

As only the DEM is adopted to calculated the normal forces in the columns, no judgements are given concerning the applicability of the 'distribution method'. However, it might be worthwhile to investigate the combination of the method described in Section 4.3.7. in the DEM method.

4.4 implementation of the genetic algorithm in the design tool

Section 4.4 deals with the utilised operators, the various general genetic algorithm parameters, and the implementation of the fitness function. The choices made in relation to the utilised genetic operators are explained in Section 4.4.1. In Section 4.4.2 the input of the general parameters for the design tool are discussed. In the final section, the design of the fitness function with its several parameters is discussed.

4.4.1 representation of the genome

A load-bearing column can be coded in a chromosome with an n-bit long string for representing different portions including the allocation of the top and bottom side, the thickness, and the activity of the column. For instance, three types, i.e. the smallest size [00], the medium size [01], and the biggest size [10], could denote the

---

[26] Vambersky [22]

portion of the thickness of the column. This already makes clear that representing the genome with bit values is not that insightful. Besides, nine times out of ten, a real number genome will perform better than a binary string genome[27]. However, actual performance depends on the problem trying to be solved.

For these two reasons, although in the previous stages examples were given with a binary representation, for the genetic algorithm on optimising the allocation of load bearing columns the real number genome representation is used.

An example of a real number genome for a single column as is used in the genetic algorithm [for the meaning of the values, see Section 4.4.2 - *number of genes per column*]:

[2,1,2,0,4,1]

It needs also to be noted that the genetic operators are closely coupled to the representation. It is possible to choose the 'right' representation but the 'wrong' genetic operator, or vice versa.

4.4.2 genetic operators

To implement a simple genetic algorithm, three basic operators are needed; selection, crossover, and mutation, discussed thoroughly in Chapter 4 of book two of this Master's thesis. Besides these three operators, other operators can be utilised as well.

initiation

Initiation requires the creation of sufficient chromosomes to fill the population. The genes of these chromosomes are set to a random value. The population then is described in a coded state. After all individuals are re-declared, the initial population can be given a little push in the 'right' direction. In the script for the design of a transfer zone, this push is given by checking the columns for maximum length and by assuring that two columns will not have the exact same centre line.

evaluation

The fitness of an individual in a GA is the value of the fitness function for its phenotype. To calculate the fitness, the chromosome must be first decoded as was stated in the previous paragraph, and then sent to the simulation model. The model will return a value indicating fitness. The various fitness parameters will be discussed in Section 4.6.

---

[27] MIT [7]

selection and elitism

Selection methods can range from *genetically conservative* (e.g. fit-fit selection, an individual is paired with its next fittest individual) to *genetically disruptive* (e.g. fit-weak selection, the fittest individual is paired with the least fit[28]. The next fittest is paired with the next least fit, etc.). The utilised roulette wheel selection is one of the traditional GA selection techniques. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness value. The roulette wheel selection is a moderately strong selection technique, since fit individuals are not guaranteed to be selected for, but have a somewhat greater chance. Based on this selection operator, two parents will crossover to form two offspring. It is implemented in the model in a way that the first chosen parent is excluded from selection so it can not crossover with itself.

Elitism is an addition to the selection operator. It can be compared with cloning and although in real life there is a lot of (social) resistance against cloning, for genetic algorithms elitism is a strong and important selection operator. In elitism, if the fitness of an individual in the previous population is larger than that of every individual in that population, this individual is preserved into the current generation. Else such individuals can be lost if they are not selected to reproduce or if they are destroyed by crossover or mutation. Introducing elitism, the best individual generated up to generation t can be included in the population at generation t + 1. Many researchers have found that elitism significantly improves the GA's performance[29].

In the GA script for the column allocation problem the fittest solution will be placed in the last position of the new population [Figure 4.12].

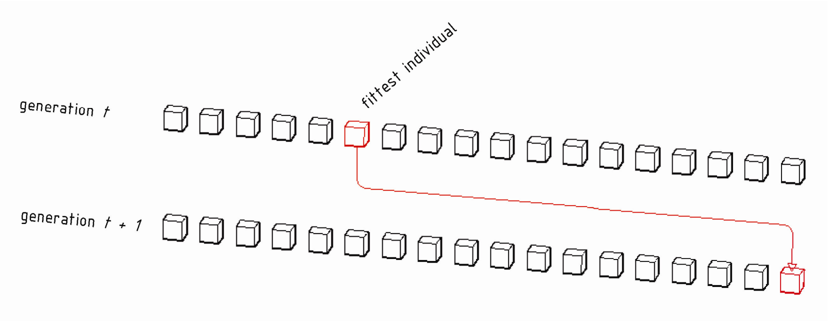


Fig. 4.12. Elitism visualised; the fittest individual is replaced in the next generation at the last position

---

[28] Bartlett [1]
[29] Mitchell [5]

crossover and mutation

These two operators are thoroughly discussed in book two of this Master's thesis. In Section 4.3.2 of book two, the crossover and mutation rate and their values are explained.

replacement[30]

Replacement is the last stage of any breeding cycle. Two parents are drawn from a fixed size population. They breed two children, but not all four can return to the population, so two of them must be replaced. The technique used to decide which individuals stay in a population and which are replaced is equal to the selection in influencing convergence.

With **weak parent replacement**, a weaker parent is replaced by a stronger child. In effect, with the four individuals only the fittest two, parent or child, return to the population. This keeps improving the overall fitness of the population when paired with a selection technique that selects both fit and weak parents for crossing, but if weak individuals don't get a chance to be selected the opportunity will never arise to replace them.

With **both parents replacement**, the child replaces the parent. This keeps the population and genetic material moving around, yet can lead to a problem when combined with a selection technique that strongly favours fit parents. The fit breed and then are disposed of. If their offspring are not as fit, the population will be degraded.
The third replacement schema replaces the **two weakest individuals** in the population with the children, as long as the children are fitter. The parents are included in the search for the weakest individuals. This technique rapidly improves the overall fitness of the population, and works well with large population where very unfit individuals would otherwise be left in for a long time.

Finally, the children replace two **randomly chosen individuals** in the population. The parents are also candidates for selection. This can be useful for prolonging the search in small populations, since weak individuals can be introduced to the population.

For the genetic algorithm dealing with the problem of allocation columns, the 'both parents' replacement method is adopted.

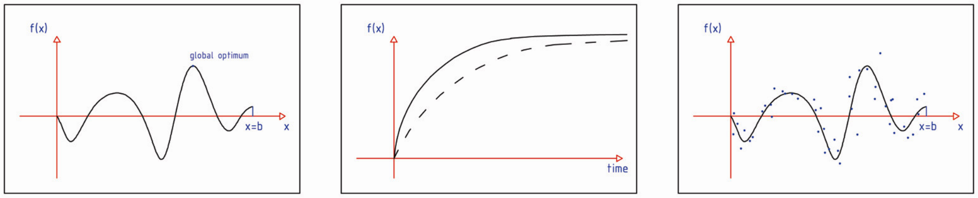
---

[30] Bartlett [1]

4.5 numerical values of the general genetic algorithm parameters

search strategies and number of generations

The search or optimisation process consists of initialising the population and then breeding new individuals until the termination condition(s) is/are met. There can be several, somewhat opposed, goals for the search process[31] [Figure 4.13], one of which is to *find the global optimum*, although this can never be assured. There is always a chance that the next iteration in the search will produce a better solution. Alternatively, the search could run for years and not produce any better solution than it did in the first ten iterations.



finding the global optimum

quick convergence

divers solutions

Fig. 4.13. Three goals for the search process

Another goal is *quick convergence*. When the object function is expensive (in time) to run, quick convergence is desirable, however, the chance of converging on a local, and possibly quite substandard optima, is increased.

The *production of a range of divers, but still good solutions* is another goal. When the solution space contains several distinct optima, which are similar in fitness, it is useful to be able to select between them, since some combinations of factor values in the model may be more feasible than others.

For the 'simple' design tool, the first two goals of the search process are of greater importance than the latter one. The total number of generations is dependant on the governing condition given by the user. Selection can be made between three conditions: the maximum number of generations, the minimum fitness that needs to be reached, or the minimum improvement of the fitness over a number of generations. For the best results, it is suggested to the latter of the three conditions.

---

[31] Bartlett [1]

population size

A population consists of a number of individuals being tested, with the phenotyping parameters defining the individuals. A fixed population size can be used, which is initialised with randomly valued individuals. There are other techniques for managing the population size, but with using a fixed size, the coding is simplified, and probably of little importance when it comes to the efficiency of getting to the end result[32].

Running a simulation with large models can be expensive in terms of time, so a smaller population may be desirable. But if the population is too small then the loss of genetic diversity may compromise the search. The other extreme, being a population size that is too large, the initiation takes too much time. Another potential problem is that the search may be flooded with an abundance of genetic diversity. In a GA, the genetic distance can be assessed in selecting parents to be selected for reproduction. If there are several groups of fit individuals clustering around different optima, the union of individuals from two distinct optima may not keep the genetically 'good' part of either of the parents. In the worst case this type of 'mixture of races', as one could call it, could depopulate the areas around the various optima. More likely is that the hill climbing around the optima will be deficient, which further increases the time cost.

number of columns

Per individual, a number of columns needs to be generated that can transfer the load from the superstructure to the substructure. Similar to the number of individuals, a larger number of columns increases the run time of the model. The difference, however, is that the population size actually is a general GA parameter, whereas the number of columns is not. The problem of allocating columns to transfer loads between two different grid systems, is the problem of minimising the number of columns. At least, with given column thicknesses as fixed parameters.

The problem with a fixed number of columns is that it cannot be assured that the (near) optimal solution was obtained. With the number of columns evolving to an optimal value including the column thicknesses, or in other words the desired slenderness of the columns, denoted by the user, the solutions generates, amongst other fitness parameters towards an state of minimal volume. In order to be able to work with various numbers of columns per run and per individual, an activity gene for every column is added to the chromosome. Initially, a large number of columns is given in the input frame by the user. Dependant on the activity value ('awake' or 'asleep') a column then participates in the determination of the fitness value of the structure or not.

---

[32] Bartlett [1]

possible positions for the columns on the grid lines

Another parameter of which the value is determined by the user of the model is the number of possible positions for the columns on the grid lines. When trying to create an image of randomness in the structure, three aspects play an important role; first, the different thicknesses of the columns, second, the location of the columns, and third, the spread of the slanting value of all columns. The latter two are greatly influenced by the number of possible positions on the grid lines. But if the number of possible positions is too large, the initiation (again) takes too much time.

number of genes per column

Six genes are needed to present the characteristics of a single column.

1. the grid line number for the lower end point of the column (bottom)
2. the position on this bottom grid line
3. the grid line number for the higher end point of the column (top)
4. the position on this top grid line
5. the thickness of the column
6. the activity of the column

So just like with the ant analogy [see Section 3.2.2 of book two]: the elements are very trivial, but the system is to be non-trivial (no solution to foreseen) and robust. This results in a 'string' length of 6 x n, with n is the number of columns per individual. In the VBA script the given values are not saved as one single string per individual, but as a 2D-array. For one complete generation, the values of all six genes are placed in one single 3D-array [Figure 4.14]:

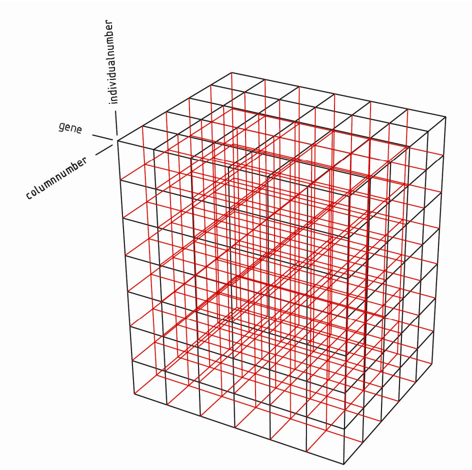
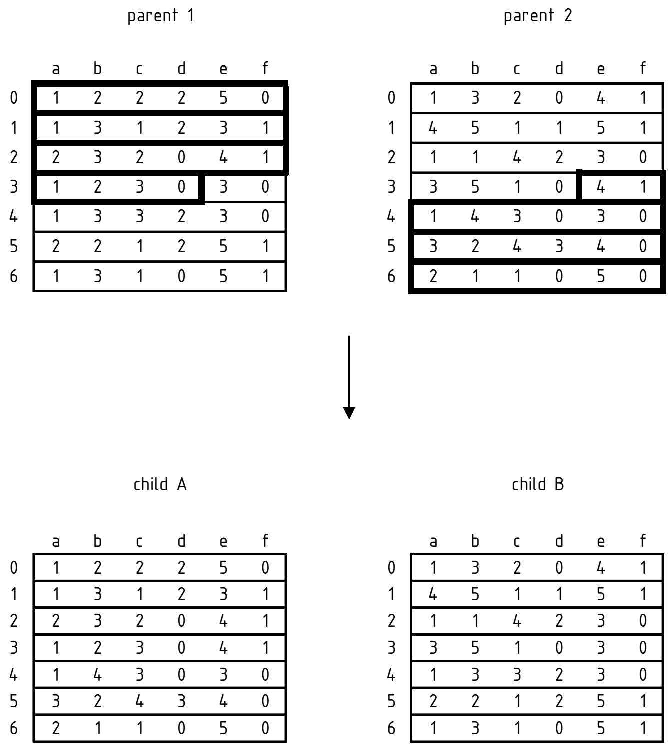


Fig. 4.14. A visualisation of the 3D-array including the individualnumber, the columnnumber and the gene for one generation

crossover rate

Crossover splices two chromosomes (two parents) at the gene level at a randomly selected crossover point. The second portion of the parents' genes is then exchanged. Figure 4.15 shows two individuals with seven (zero to six) columns and six genes (a to f) crossover after the twenty-second gene.



parent 1

	a	b	c	d	e	f
0	1	2	2	2	5	0
1	1	3	1	2	3	1
2	2	3	2	0	4	1
3	1	2	3	0	3	0
4	1	3	3	2	3	0
5	2	2	1	2	5	1
6	1	3	1	0	5	1

parent 2

	a	b	c	d	e	f
0	1	3	2	0	4	1
1	4	5	1	1	5	1
2	1	1	4	2	3	0
3	3	5	1	0	4	1
4	1	4	3	0	3	0
5	3	2	4	3	4	0
6	2	1	1	0	5	0

child A

	a	b	c	d	e	f
0	1	2	2	2	5	0
1	1	3	1	2	3	1
2	2	3	2	0	4	1
3	1	2	3	0	4	1
4	1	4	3	0	3	0
5	3	2	4	3	4	0
6	2	1	1	0	5	0

child B

	a	b	c	d	e	f
0	1	3	2	0	4	1
1	4	5	1	1	5	1
2	1	1	4	2	3	0
3	3	5	1	0	3	0
4	1	3	3	2	3	0
5	2	2	1	2	5	1
6	1	3	1	0	5	1

[0…6] - column number
[a…f] - gene number

Fig. 4.15. One point crossover, where child A and child B are offspring of parent 1 and parent 2

It is possible for the slice point to be zero, or to be equal to the number of genes in the 'string', in which case a straight copy is the result, and thus creating a child A that is equivalent to parent 1 and child B is equivalent to parent 2. It needs to be noted that the effect of the crossover for the fitness of the chromosome depends largely on where the crossover point is located[33].

---

[33] Bartlett [1]

mutation rate

The mutation process is the process of randomly disturbing genetic information. Mutation operates at the gene level. A mechanism similar to tossing a coin is employed, meaning that if a random number between 0 and 1 is less then the mutation rate (usually a quite small value) the gene value is changed into a randomly value. As mutation provides an 'insurance policy' against fixation of the same value at a random gene for every individual, it is suggested to increase the mutation rate through the generations, that is, if the number of (almost) identical individuals is large.

4.6 parameters for the fitness function of the design tool

4.6.1 functions of the fitness function

The fitness function is possibly the most important component of a GA. The purpose of the fitness function is to map a chromosome representation into a scalar value[34].

Since each chromosome represents a potential solution, the evaluation of the fitness function quantifies the quality of that chromosome. Next to this, selection operators make use of the fitness evaluation of chromosomes.

It is therefore extremely important that the fitness function accurately models the optimisation problem. The fitness function should include all criteria to be optimised. In addition to optimisation criteria, the fitness function can also reflect the constraints of the problem through penalisation of those individuals that violate constraints. However, it is not required that the constraints are encapsulated within the fitness function; constraints can also be incorporated in the initialisation, reproduction, and mutation operators.

The fitness function can be thought of as some measure of profit that needs to be maximised[35]. Or in other words, fitness is the probability that an organism will live to reproduce. But for it to actually mean anything, several fitness parameters need to be combined to form this function. Three important aspects then have to be dealt with. First, the parameters, in someway, need to be contradictory, meaning that the increase of one parameter will result in the decrease of another parameter. Second, a weight should be given to every parameter to emphasize the influence of it on the final fitness value. Third, the representation of the fitness values needs to be in a format that the mutual values can be compared.

---

[34] Engelbrecht [2]

[35] Bartlett [1]

In combination with the weighing factors, the values of the fitness parameters can then be easily summed to a total fitness for the individual. Therefore, the fitness parameters will have values between zero and one. Where zero, or better a value very close to zero means a bad fitness, or a non-possible solution and a value of one means a 'perfect' solution for the concerning criterion.

4.6.2 the insertion of parameters in the fitness function

Below the criteria, or in other words, the parameters of the fitness function to be optimised (meaning either maximised or minimised) are given.

1. the overall stability of the structure, including the capacity to take the horizontal loads and resulting moments;
2. the load bearing capacity of the columns in relation with the actual normal load;
3. the total volume of the structural elements;
4. the summed angle of all columns;
5. the intersecting volume of the columns, or the minimal distance between columns;
6. and the variation of the possible gene values to create a feeling of randomness.

Two parameters of these criteria have to be answered fully; it needs to be guaranteed that the stability, and the horizontal and vertical loads, including the resulting moments can be transferred by the columns from the super- to the substructure. This means that if the model detects that a solution is not stable or can not transfer all loads, the fitness of the concerning individual is (close to) zero, e.g. 1.10^{-12}. In case an individual is stable and can transfer all loads, the value of stability fitness parameters is one, and of the bearing capacity fitness parameter is between zero and one. In other words, the value for criterion 1 is discrete and for criterion 2 real fitness values will be used, as it is useful to take into account the relation between the bearing capacity of the columns including buckling and bearing capacity of the columns without buckling, in order to utilise the full capacity of the material.

The value of the total volume of the structural elements needs to be inverted as this criterion needs to be minimised. The same holds true for the summed angle of all columns and the intersecting volume of the columns.

4.6.3 contradiction of the fitness parameters

As mentioned before, the effects the different values of the fitness criteria have on the structure need to be contradictorily. This will have to find its consequences in the values of the six genes of a single column that represent three aspects of the column.

1. the length and/or angle (gene 1 to 4)
2. the thickness (gene 5)
3. the activity (gene 6)

Between the six criteria given in Section 4.6.2 the following contradictions, based on two column aspects, occur, regardless of the number of columns.

1. if the summed column length of all columns slides to a minimum value, obviously, the total volume of the load bearing elements will become smaller as well. This means that the volume fitness parameter reaches the optimal value, as the fitness parameter is reversely proportional with the actual total volume of the columns. Two other fitness parameters will increase as well when the summed column length is as short as possible. First, as the buckling force of columns is dependant on the column length, the maximal load bearing capacity of a single column is smaller when dealing with a slender column (or a longer column with a fixed thickness). Second, the intersecting volume parameter is more likely to be optimised when shorter columns are used.
   Naturally, one or more parameters must contradict the movement towards shorter columns. One of them is the stability fitness parameter. This because, if the length of all columns reaches the minimum value, all columns are vertical. In this situation, no horizontal loads can be transferred from the superstructure to the substructure.

2. considering the thickness of the columns, the optimisation of the total volume parameter and the total intersecting parameter demands slim columns. Opposite to this, the fitness parameter for the total vertical load bearing capacity calls for thick columns if a small number of column is active. The variation fitness parameter also prevents all columns being of a small thickness.

When the contradictions are regarded from the fitness parameters point of view, instead of the column genes point of view, the following enumeration, based on a fixed number of columns can be made.

- For the optimal fitness value of the overall stability of the structure, the columns need to slant to be able to transfer the horizontal load, and need to be as thick as possible.
- For the optimal fitness value of the total load bearing capacity of the columns, the columns need to be as short and thick as possible, in other words the slenderness needs to be as low as possible.
- For the optimal fitness value of the total volume of the structural elements, the columns need to be as short and slim as possible.
- For the optimal fitness value of the summed angle of all columns with the vertical, the gene values will not be favoured in anyway.
- For the optimal fitness value of the intersecting volume of the columns, or the minimal distance between columns, the columns, similar to the total volume criterion, need to be as short and slim as possible.
- For the optimal fitness value of the spread of the possible gene values to create a feeling of randomness, the columns need to have a small, medium and large diameter and have several different lengths.

The optimal solution, e.g. the fittest individual after a run, will have a fitness value between zero and six, when the weighing factors are between zero and one. Although the fitness value can never reach the maximum of six due to the contradictory criteria, this value can easily be used to judge the overall fitness of the individuals.

4.7 concluding remarks

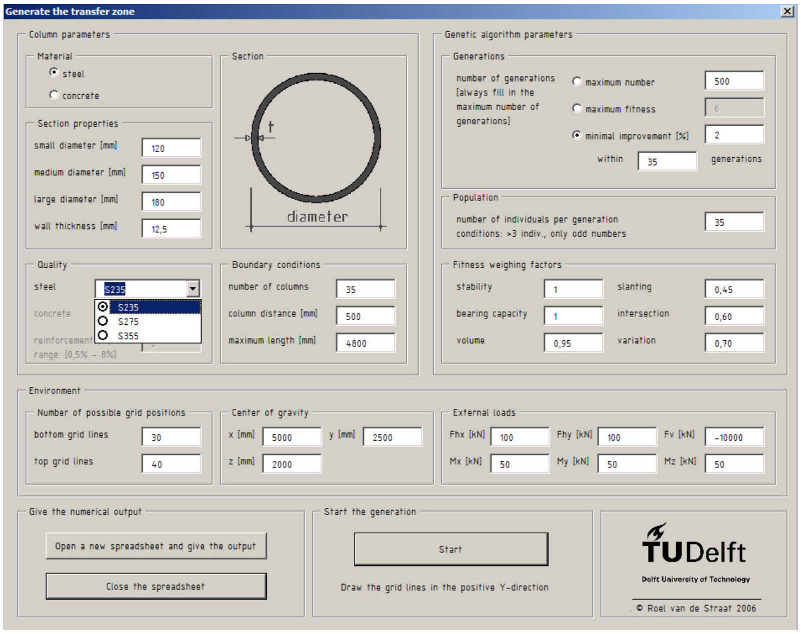
Considering the large number of parameters (the number of columns, the possible locations, and the possible thicknesses) and characteristics (the grid system, the seemingly randomness, and the demand for an optimal solution) of the problem, the genetic algorithm method can be well implemented in the design tool to achieve good results. Another aspect is, that GA's are well discussed in the literature and therefore the basic aspects of the implementation in a script are facilitated.
In this chapter, it is shown that numerous structural aspects play an important role in the validation of the design tool [see also Chapter 6]. And together with the numerical values of the general genetic algorithm parameters and the utilised parameters for the fitness function, the onset for scripting the structural optimisation tool is given.

5 scripting the structural optimisation tool

Chapter 5 deals with the scripting and scripting aspects of the design optimisation tool. Section 5.1 presents the flowchart of the VBA script, which reveals the way the different procedures are called upon. In the succeeding sections, different aspects and procedures of the script are clarified.

5.1 utilising the VBA script to design the transfer zone

One of the goals for the completed design of the VBA script is to create a user interface where the user can give values for the several operator parameters, the geometrical outline, the various forces, etc. The input can be given numerically in the graphical user interface (GUI) [Figure 5.1], or the command line of AutoCAD or graphically in the Modelspace of AutoCAD.



● steel
○ concrete

small diameter [mm]: 120
medium diameter [mm]: 150
large diameter [mm]: 180
wall thickness [mm]: 12,5

steel: S235
● S235
○ S275
○ S355
concrete
reinforcement range: [0,5% - 8%]

number of columns: 35
column distance [mm]: 500
maximum length [mm]: 4800

number of generations [always fill in the maximum number of generations]
○ maximum number: 500
○ maximum fitness: 6
● minimal improvement [%]: 2
within 35 generations

number of individuals per generation conditions: >3 indiv., only odd numbers: 35

stability: 1
bearing capacity: 1
volume: 0,95
slanting: 0,45
intersection: 0,60
variation: 0,70

bottom grid lines: 30
top grid lines: 40

x [mm]: 5000
y [mm]: 2500
z [mm]: 2000

Fhx [kN]: 100
Fhy [kN]: 100
Fv [kN]: -10000
Mx [kN]: 50
My [kN]: 50
Mz [kN]: 50

Draw the grid lines in the positive Y-direction

Fig. 5.1. The GUI of the design optimisation tool

The flowchart of the VBA script is shown below [see also Appendix B for a larger figure]. The main program is indicated by the Main Subs, the column under the Start 'button'. These subs are supported by Sub's Subs (second column) and the Functions (third column).

In comparison with the general flowchart of genetic algorithms in Figure 4.7 of book two, the onset of the evolutionary process is given by the input of general parameters, the creating of the AutoCAD drawing layers, and the graphical determination of the grid lines in the AutoCAD environment.

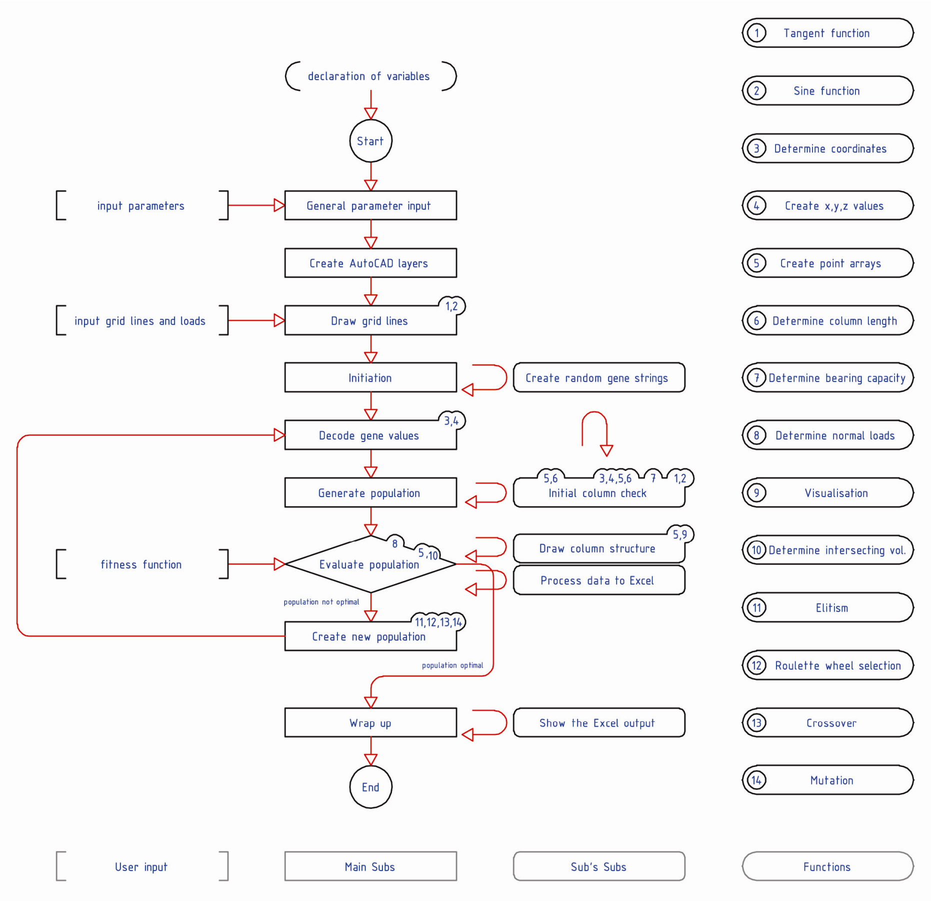


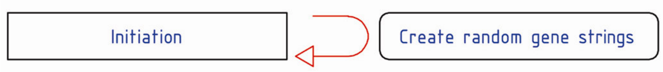
Fig. 5.2. The flowchart of the VBA script of the design tool

The other procedures in the simple GA flowchart can easily be distilled from the flowchart in Figure 5.2, where the selection, crossover, and mutation operators are denoted as functions in the Main Sub 'Create new population'.

5.2 explaining the script based on the VBA script flowchart

After the determination of the general parameters (mainly via the GUI) and input of the structural grids in the AutoCAD Modelspace, several procedures are called to perform the genetic evolutionary process. In this section the four most important (sets of) Subs and Functions are explained. The Functions used per Sub are denoted with a number (i).

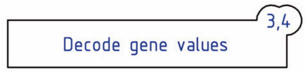
5.2.1 creation of random gene strings



In order to create the first generation – Generation 0 – the Initiation Sub calls the 'Create random gene strings' Sub. Herein for the six genes of every column of every individual 'random' but possible, values are attributed with the random number generator of VBA.

The values are drawn up in the coded variant, meaning that the locations of the top and bottom point of a column are indicated with the grid line number and the position of the column on this grid line. The thickness value is indicated with a zero, a one, or a two for respectively a small, medium, and large diameter for the column thickness, and the activity gene is valued with a zero (asleep) or a one (awake).

5.2.2 hatching of the coded gene values



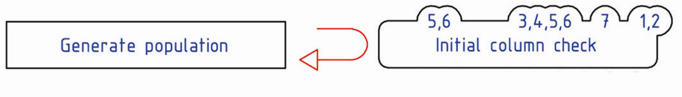
As the coded gene values cannot be used for the determination of column parameters such as the column length and angle, the values need to be decoded (3,4) – a process similar to 'hatching'[36] of an egg; the genetic properties of the creature of individual remains the same, but it is 'applicable' after hatching. This aspect of the script is based on the openStrategy Form Finding method[37]. After decoding the genome, the location of the top and bottom point of a column are given in x, y, z – coordinates, and the thickness in the diameter of the column in millimetres. After a new generation of individuals is created [Section 5.2.5] new chromosomes with coded values are send to the 'Decode gene values' Sub.

---

[36] Coenders [4]

[37] Coenders [4]

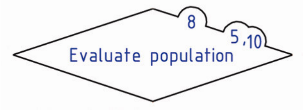
5.2.3 checking the generated individuals and columns



The initial column check calls numerous functions, and can call the 'Create random gene strings' Sub as well. After the gene values are decoded, the length of every column of every individual can be determined (5,6) and it is checked whether any of the columns have identical centre lines. If the latter is the case, the concerning column is removed and replaced by a new one with 'random' gene values in the 'Create random gene strings' Sub. Subsequently, the new coded gene values need to be decoded again (3,4) and the new column can be checked for the length (5,6).

Every column is then subjected to a routine in which the bearing capacity (7) [Section 4.3.5] and the angles with the horizontal surface and the vertical z-axis are determined (1,2).

5.2.4 evaluating the generations



After a generation of individuals is created, every individual is subjected to a fitness check. However, not before the normal column loads based on the given external loads and the location of the columns of every individual are determined.

calculation of the column normal force (8)

In order to take in to account the axial stiffness of the columns, a Finite Element Method is adopted, called the Discrete Element Method (DEM) [J.W. Welleman, October 2006]. This method was used and validated for a 2D-system, but had to be adapted for a 3D-environment. This means that the structure is transferred from a system with three degrees of freedom $\{u\}^T = (u \; v \; \varphi)$ to a system with six degrees of freedom $\{u\}^T = (u \; v \; w \; \varphi_x \; \varphi_y \; \varphi_z)$ in the centre of gravity of the superstructure. The assumption needs to be made that the superstructure is a rigid body.

To determine the column normal force in the pinned columns, the equation

$$N_i = D_i \times B_i \times \{u\} \tag{5.1}$$

needs to be solved. With D_i is the constitutive matrix ($=EA_i/l_i$) and B_i is the kinematical matrix of the concerning column. The six degrees of freedom in {u} need to be derived in order to solve equation 5.1.

The system that needs to be solved for determining the unknown displacements and rotations is

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} & k_{04} & k_{05} \\ k_{10} & k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{20} & k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{30} & k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \\ k_{40} & k_{41} & k_{42} & k_{43} & k_{44} & k_{45} \\ k_{50} & k_{51} & k_{52} & k_{53} & k_{54} & k_{55} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ \varphi_x \\ \varphi_y \\ \varphi_z \end{bmatrix} \qquad 5.2$$

For the determination of the column kinematical matrix the 'unity direction vector' *l* needs to be determined. The elongation of the column, dependant only on the displacements of x_{top}, y_{top}, and z_{top} (as x_{bottom}, y_{bottom}, and z_{bottom} are fixed), can then be expressed as the product of the kinematical matrix B_i and the column vector with the degrees of freedom {u}.

$$l = \frac{\begin{pmatrix} x_{top} - x_{bottom} \\ y_{top} - y_{bottom} \\ z_{top} - z_{bottom} \end{pmatrix}}{\sqrt{(x_{top} - x_{bottom})^2 + (y_{top} - y_{bottom})^2 + (z_{top} - z_{bottom})^2}} = \frac{\begin{pmatrix} x_{top} - x_{bottom} \\ y_{top} - y_{bottom} \\ z_{top} - z_{bottom} \end{pmatrix}}{L} = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix}$$

with $|l| = 1{,}0$ ...5.3

The kinematical matrix can then be written as

$$B_i = [l_x \quad l_y \quad l_z \quad \Delta y l_z - \Delta z l_y \quad -\Delta x l_z + \Delta z l_x \quad \Delta x l_y - \Delta y l_x] \qquad 5.4$$

The next step is to determine the column stiffness matrix K_i, which is a 6 x 6 matrix that can be constructed with the following formula based on a global coordinate system [Fgure 5.3 on the next page].

$$K_i = B_i^T \times D_i \times B_i \qquad 5.5$$

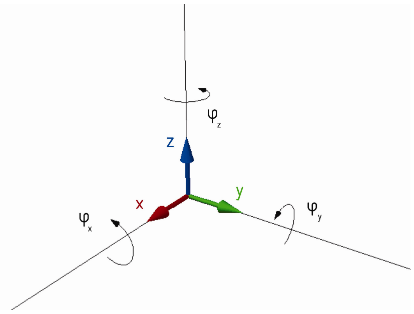


Fig. 5.3. The global coordinate system, including the positive rotations

The system stiffness matrix can be obtained by summating the stiffness matrix for every column.

$$K_{sys} = \sum_{i=0}^{i=n} K_i \tag{5.6}$$

with n is the number of columns minus 1.

In the VBA-script, the system of Equation 5.2 is solved with a Gauss-Jordan Elimination method, a method for solving a linear system of equations, where the system matrix needs not to be written in a (reduced) echelon form. This method is written in the VBA-script based on the five steps shown below. No predefined library is used to implement the method in the process of determining the degrees of freedom.

Five steps are needed to determine φ_z en from that point back substitution makes it possible to determine the other degrees of freedom. In every step, 'shadow equations' are obtained with zeros on the first non-zero column. This method is clarified below for the following 2D-system.

$$\begin{array}{ccccccc} 10 & = & 5x & + & 2y & + & z \\ 5 & = & x & + & y & + & 2z \\ 0 & = & x & + & y & + & 4z \end{array}$$

And as was already mentioned, after obtaining the degrees of freedom for the complete system, the normal load per column can be derived with equation 5.1.

matrix notation

$$\begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{matrix} (0) \\ (1) \\ (2) \end{matrix}$$

step 0 – multiply equation (1) and (2) with 5 and subtract them from equation (0)

$$\begin{array}{rcrcrcrl} 10 & = & 5x & + & 2y & + & z & \\ 25 & = & 5x & + & 5y & + & 10z & - \\ \hline -15 & = & & & -3y & + & -9z & \end{array}$$

and

$$\begin{array}{rcrcrcrl} 10 & = & 5x & + & 2y & + & z & \\ 0 & = & 5x & + & 5y & + & 20z & - \\ \hline 10 & = & & & -3y & - & 19z & \end{array}$$

matrix notation

$$\begin{bmatrix} -15 \\ 10 \end{bmatrix} = \begin{bmatrix} 0 & -3 & -9 \\ 0 & -3 & -19 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{matrix} (3) \\ (4) \end{matrix}$$

step 1 – multiply equation (4) with 1 and subtract it from equation (3)

$$\begin{array}{rcrcrl} -15 & = & -3y & + & -9z & \\ 10 & = & -3y & + & -19z & - \\ \hline -25 & = & & & 10z & \end{array}$$

matrix notation

$$\begin{bmatrix} -25 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} (5)$$

The z-value can easily be calculated, z = -2,5. Via back substitution with equation (3) and (0) the y- and x-value can be derived as well.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2,\ 5 \\ 12,\ 5 \\ -2,\ 5 \end{bmatrix}$$

As the determination of the column loads, and thus the determination of the equilibrium of the forces and stability of the equilibrium are based on this method, the VBA script only has to prevent the running of the 'normal column load' Sub in case a structural mechanism [Figure 4.10] occurs.

implementation of the fitness function

In order to determine the intersection volume fitness parameter (10), the structure needs to be drawn in AutoCAD in the 'Draw column structure' Sub. In this Sub, also the AutoCAD visualisation of the fittest individual is saved.
In the 'Evaluate population' Sub itself, the values for the different fitness parameters (4.6.2) are determined and the maximum, average, and minimum fitness are calculated and the values are then saved in MS Excel in the 'Process data to Excel' Sub.

Finally, it is checked whether the fitness values comply with any of the stopping criteria denoted by the user. If not, then a new population needs to be created [Section 5.2.5], else the script follows it path with the 'Wrap up' Sub, where all data for the run can be shown.

5.2.5 creating the new population and utilisation of the genetic operators

The 'Create new population' Sub calls four genetic operators. Via the 'Elitsm' Function (11), the fittest individuals are copied to the end position in the individual positions. The other individuals are created with the crossover procedure. With the roulette wheel selection method, two individuals (parents) are selected for 'breeding'. Every two parents create two children, two new individuals. Finally, every new chromosome, except the fittest one, is subjected to the 'Mutation' Function, resulting in a relatively small distortion of the new population.

For the first generations, it is desired for the mutation rate to be relatively high. As the number of generations increases, the mutation rate preferably decreases exponentially. The main advantage of this, is the fact that in initial large mutation rate ensures that a large search space is covered, while the distorting effect of mutation becomes smaller when individuals start to converge towards the optimal solution.

Various functions can be adopted to determine the mutation rate for every generation [Figure 5.4]. Similar to the speed of the radioactive decrease of a radioisotope, a function including a halving time parameter is chosen for describing the mutation rate.

$$p_m = \frac{p_m(0)}{2^{\frac{1}{h_{1/2}}t}} \tag{5.7}$$

In Equation 5.7, $p_m(0)$ is the mutation rate for the first generation (t = 0), a value in general smaller than 0,1. The halving time, $h_{1/2}$, is the number of generations needed to halve the mutation rate over that number of generations.

Example – red line in Figure 5.4

*Suppose that the start value (*p(0)*) is 1 and the halving time (*$h_{1/2}$*) is 2. For* t = *0 the mutation rate is equal to* p(0). *After two generations the mutation rate is halved,* p(2) = *0,5. Another two generations, and the mutation rate is halve again,* p(4) = *0,25 etc.*

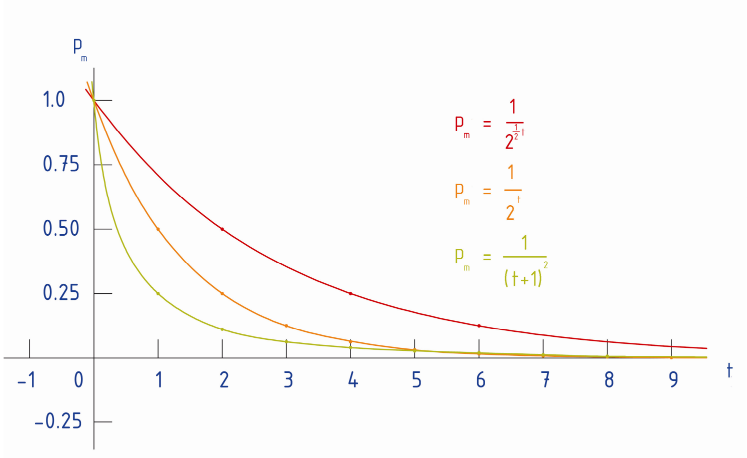


Fig. 5.4. Exponential decreasing y-values for increasing x-values for three different functions

After the new population has been created, the children follow the same path to the 'Evaluate population' Sub their parents went as from the 'Decode gene values' Sub.

5.3 concluding remarks

One of the main features of this Master's thesis is the flowchart of the VBA script of the design tool. Herein, the VBA procedures (Main Subs, Sub's Subs, and Functions) are presented that need to be called upon to create the optimal solution. The genetic evolutionary processes are denoted in the Main Subs, and are supported by different Sub's Sub, such as an initial column check and a procedure for drawing the structures in AutoCAD.

In order to be able to generate a first set of individuals (generation 0), a random gene operator is called, which fills the chromosome of every individual with possible gene values in a coded set. After decoding these values, the column characteristics, such as length and angle can be determined. In the 'Evaluate population' Sub, the structures are evaluated on their fitness based on the fitness function [see Section 4.6]. Here, also the Function 'Determine normal loads' is utilised to calculate the normal load in the active columns of every structure. This calculation is performed with the Discrete Element Method, where loads and moments in the main axial directions in the centre of gravity of a rigid body are transferred to pin-connected columns based on their location, length, and axial stiffness. To finalise the exposé of the flowchart, the 'Create new population' sub, including the determination of the variable mutation rate, is discussed.

6 validating the structural optimisation tool

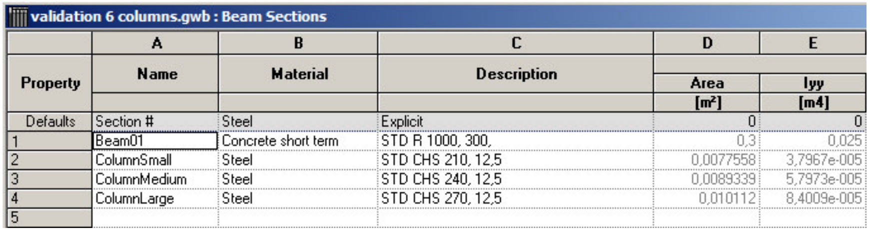
Various parameters for the several genetic algorithm operators and the fitness function need to be valued adequately in order to drive the evolutionary process to an optimal solution. In this chapter, three aspects of main interest are tested and validated for their contribution in estimating the fittest solution.

6.1 calculation of the normal column loads with the DEM method

In Section 5.2.4, the DEM method is illustrated as a method for the calculation of the normal loads of a column structure transferring load from a rigid body to a rigid foundation. Here the outcome of a computational run is tested by comparing the results of the VBA script calculations with results from the structural analysis tool Oasys GSA.

6.1.1 normal column loads for a statically definite structure

A random statically definite column structure (with six columns in a 3D environment) is set up in an Oasys GSA model. The columns, with different cross sections [Figure 6.1] are pin-connected with the foundation and the superstructure, which consists of numerous large concrete beams, which are rigidly connected (with the properties of Beam01 of Figure 6.1].



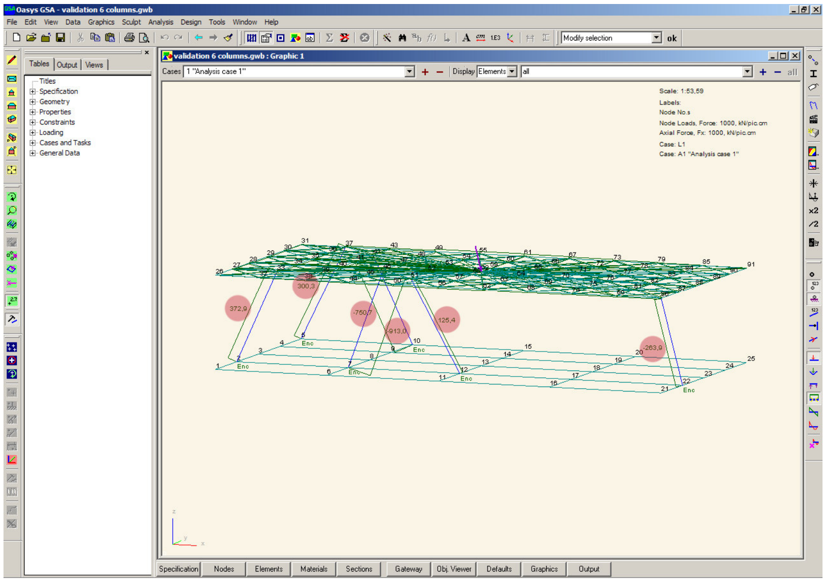
validation 6 columns.gwb : Beam Sections

	A	B	C	D	E
Property	Name	Material	Description	Area [m²]	Iyy [m4]
Defaults	Section #	Steel	Explicit	0	0
1	Beam01	Concrete short term	STD R 1000, 300,	0,3	0,025
2	ColumnSmall	Steel	STD CHS 210, 12,5	0,0077558	3,7967e-005
3	ColumnMedium	Steel	STD CHS 240, 12,5	0,0089339	5,7973e-005
4	ColumnLarge	Steel	STD CHS 270, 12,5	0,010112	8,4009e-005
5					

Fig. 6.1. Three circular hollow steel sections for the columns of the transfer zone; thickness: [210, 240, or 270 mm, wall thickness: 12,5 mm]

By adding three arbitrary node loads (F_x = 200 kN, F_y = 100 kN, and F_z = -1000 kN) a calculation could be made of which the results are [see also Figure 6.2.]

N(0) = 379,2
N(1) = 300,3
N (2) = -913,0
N (3) = 125,4
N (4) = -750,7
N (5) = -263,9



Scale: 1:53,59
Labels:
Node No.s
Node Loads, Force: 1000, kN/pic.cm
Axial Force, Fx: 1000, kN/pic.cm
Case: L1
Case: A1 "Analysis case 1"

Fig. 6.2. Calculated column loads in Oasys GSA for a statically definite structure

The same structure is manually set up in the VBA script and drawn in the AutoCAD environment [Figure 6.3], and the exact same results as derived in the structural analysis tool were calculated with the DEM method [Figure 6.4].

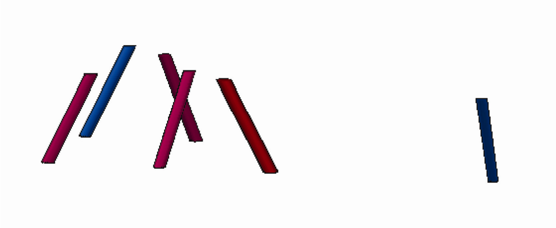


Fig. 6.3. Six columns in the AutoCAD environment

		column 0	column 1	column 2	column 3	column 4	column 5
generation 0	individual 0	372,8571429	300,2719855	-913,0141517	125,3571429	-750,6695424	-263,9268531

Fig. 6.4. Results from the DEM method in the VBA script

6.1.2 normal column loads for a statically indefinite structure

To the structure presented in Section 6.1.2, two columns are added, and thus a statically indefinite structure is modelled in Oasys GSA and VBA [Figure 6.5]. Again, results from both tools are compared to validate the DEM method.

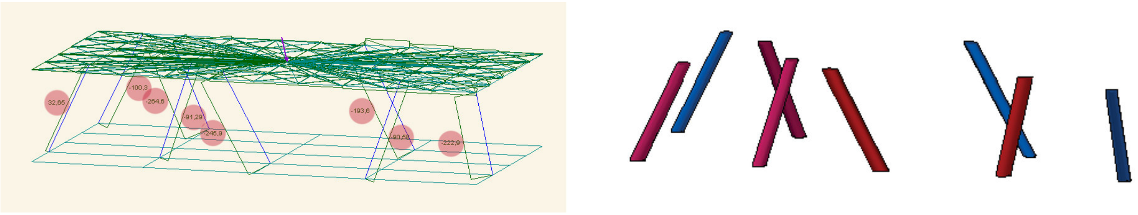


Fig. 6.5. The statically indefinite structure with 8 columns in GSA and AutoCAD

N (0) = 32,65
N (1) = -100,3
N (2) = -91,29
N (3) = -264,6
N (4) = -245,9
N (5) = -193,6
N (6) = -90,58
N (7) = -222,9

		column 0	column 1	column 2	column 3	column 4	column 5	column 6	column 7
generation 0	individual 0	30,01717755	-97,07505292	-86,12694687	-269,6384298	-226,3136526	-155,9427939	-143,7338255	-214,4473363

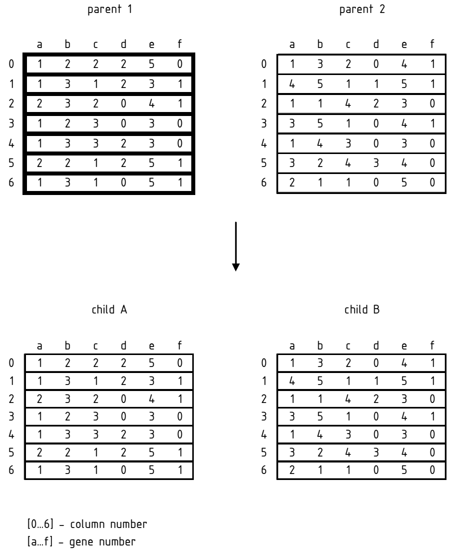
Fig. 6.6. Calculated column loads of Oasys GSA (top) and the DEM method in the VBA script

Figure 6.6 shows that with the exception of columns 5 and 6, the difference between the results from Oasys GSA and the DEM method in the VBA script stays within the 10%. The difference between the two calculations arises as the superstructure of the GSA model distributes the loads differently than the assumed rigid body, as can be seen in Figure 6.5 where normal loads in the beams of the superstructure are shown.

6.2 determination of the GA operator values

6.2.1 influence of the crossover and mutation operator

With a crossover rate of zero, the point of crossover is set at the 'string length', which is the location after the last gene in the 2D-gene matrix of an individual. Every gene of parent 1 prior to the point of crossover, i.e. all genes of the individual, is copied to the empty gene locations of child A. As a result, child A is an exact copy of parent 1. The same holds true for child B and parent 2, see Figure 6.7, next page.



parent 1

	a	b	c	d	e	f
0	1	2	2	2	5	0
1	1	3	1	2	3	1
2	2	3	2	0	4	1
3	1	2	3	0	3	0
4	1	3	3	2	3	0
5	2	2	1	2	5	1
6	1	3	1	0	5	1

parent 2

	a	b	c	d	e	f
0	1	3	2	0	4	1
1	4	5	1	1	5	1
2	1	1	4	2	3	0
3	3	5	1	0	4	1
4	1	4	3	0	3	0
5	3	2	4	3	4	0
6	2	1	1	0	5	0

child A

	a	b	c	d	e	f
0	1	2	2	2	5	0
1	1	3	1	2	3	1
2	2	3	2	0	4	1
3	1	2	3	0	3	0
4	1	3	3	2	3	0
5	2	2	1	2	5	1
6	1	3	1	0	5	1

child B

	a	b	c	d	e	f
0	1	3	2	0	4	1
1	4	5	1	1	5	1
2	1	1	4	2	3	0
3	3	5	1	0	4	1
4	1	4	3	0	3	0
5	3	2	4	3	4	0
6	2	1	1	0	5	0

[0...6] – column number
[a...f] – gene number

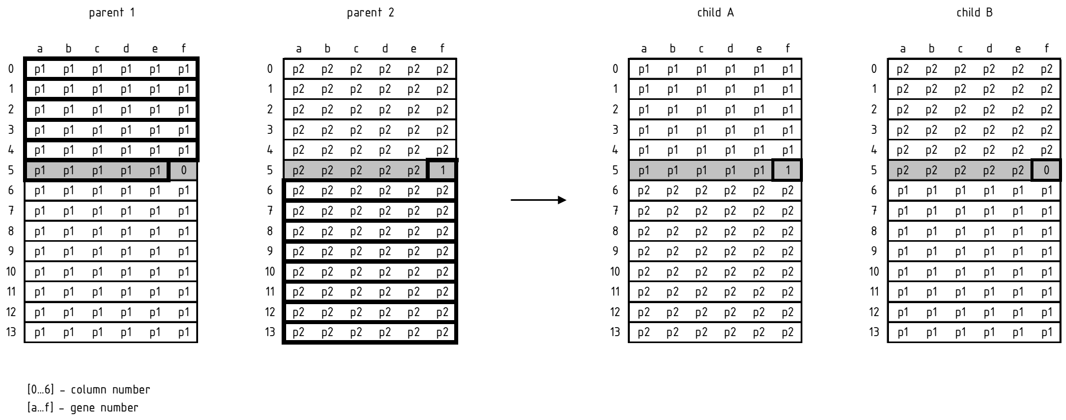
Fig. 6.7. One point crossover between parent 1 and parent 2 with the point of crossover at 'string length'.

These new individuals, child A and B, will not be altered in case of a mutation rate of zero, and as the selection of individuals for mating is biased in favour of fit individuals, the fitter individuals will again be used to create a new generation. Logically, this will result in a run where the final generation (under the restriction that enough generations where created) will consist of individuals exactly equal to the fittest individual in generation 0.

When the crossover rate is set to a larger value than '0' (not larger than '1'), all new individuals are created by mating of two parents for the previous generation. To what degree the child resembles parent 1 or parent 2 is dependant of the randomly determined crossover point.

Example

The total number of columns per individual is 14 and the crossover point is set to 35, which has as a result that the genes of column number 5 of both parents are partly copied to child A and partly to child B. In Figure 6.8 and 6.9, it can be seen that the genes of columns 0 to 4 are copied from parent 1 to child A and from parent 2 to child B. The characteristics of columns 6 to 13 are transferred from parent 1 to child B and from parent 2 to child A. For column number 5 of child A the location (gene 'a' to 'd') and the thickness (gene 'e') are similar to those values of parent 1. The difference between column number 5 of parent 1 and child A lies in the activity value. The column 'inherited' the 'awake' state from parent 2, whereas column number 5 of child B is 'asleep'.



parent 1

	a	b	c	d	e	f
0	p1	p1	p1	p1	p1	p1
1	p1	p1	p1	p1	p1	p1
2	p1	p1	p1	p1	p1	p1
3	p1	p1	p1	p1	p1	p1
4	p1	p1	p1	p1	p1	p1
5	p1	p1	p1	p1	p1	0
6	p1	p1	p1	p1	p1	p1
7	p1	p1	p1	p1	p1	p1
8	p1	p1	p1	p1	p1	p1
9	p1	p1	p1	p1	p1	p1
10	p1	p1	p1	p1	p1	p1
11	p1	p1	p1	p1	p1	p1
12	p1	p1	p1	p1	p1	p1
13	p1	p1	p1	p1	p1	p1

parent 2

	a	b	c	d	e	f
0	p2	p2	p2	p2	p2	p2
1	p2	p2	p2	p2	p2	p2
2	p2	p2	p2	p2	p2	p2
3	p2	p2	p2	p2	p2	p2
4	p2	p2	p2	p2	p2	p2
5	p2	p2	p2	p2	p2	1
6	p2	p2	p2	p2	p2	p2
7	p2	p2	p2	p2	p2	p2
8	p2	p2	p2	p2	p2	p2
9	p2	p2	p2	p2	p2	p2
10	p2	p2	p2	p2	p2	p2
11	p2	p2	p2	p2	p2	p2
12	p2	p2	p2	p2	p2	p2
13	p2	p2	p2	p2	p2	p2

child A

	a	b	c	d	e	f
0	p1	p1	p1	p1	p1	p1
1	p1	p1	p1	p1	p1	p1
2	p1	p1	p1	p1	p1	p1
3	p1	p1	p1	p1	p1	p1
4	p1	p1	p1	p1	p1	p1
5	p1	p1	p1	p1	p1	1
6	p2	p2	p2	p2	p2	p2
7	p2	p2	p2	p2	p2	p2
8	p2	p2	p2	p2	p2	p2
9	p2	p2	p2	p2	p2	p2
10	p2	p2	p2	p2	p2	p2
11	p2	p2	p2	p2	p2	p2
12	p2	p2	p2	p2	p2	p2
13	p2	p2	p2	p2	p2	p2

child B

	a	b	c	d	e	f
0	p2	p2	p2	p2	p2	p2
1	p2	p2	p2	p2	p2	p2
2	p2	p2	p2	p2	p2	p2
3	p2	p2	p2	p2	p2	p2
4	p2	p2	p2	p2	p2	p2
5	p2	p2	p2	p2	p2	0
6	p1	p1	p1	p1	p1	p1
7	p1	p1	p1	p1	p1	p1
8	p1	p1	p1	p1	p1	p1
9	p1	p1	p1	p1	p1	p1
10	p1	p1	p1	p1	p1	p1
11	p1	p1	p1	p1	p1	p1
12	p1	p1	p1	p1	p1	p1
13	p1	p1	p1	p1	p1	p1

[0...6] - column number
[a...f] - gene number

Fig. 6.8. One point crossover between parent 1 and parent 2 with the point of crossover at gene location 35.

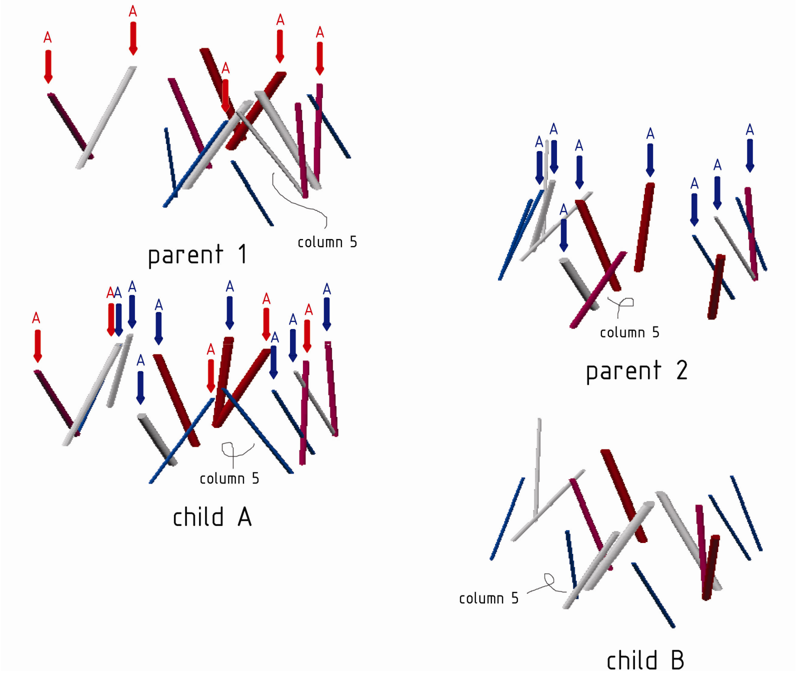


Fig. 6.9 Child A 'inherited' the characteristics of column number 0 to 4 from parent 1 and column number 6 to 13 from parent 2 (column numbers not given in the figure).

When also a positive non-zero value (not larger than '1') for the mutation rate is set, the new individuals can be subjected to mutation, meaning that with a certain probability, gene values can be altered, as shown in Figure 6.10. The figure shows an example with only active columns.

As can be seen in Figure 6.10. the crossover point is located in the gene string of column 4. The columns indicated with filled circles are copied from parents 1 (red) and 2 (blue) to child A. The open circles correspond with column numbers that are copied to child B. In this case, the combination of the gene values of column 4 of parent 1 with the gene values of column 4 of parent 2 resulted in columns not in accordance with the restriction of the maximum length. Consequently, a new column 4 for both child A and B is created by calling the random gene appointing function in the VBA script.

After crossover, the mutation function is called, which resulted in the mutation of two columns; number 11 of child A and number 9 of child B. Column number 11 of child A is copied from parent 2, but as the grid line location values are mutated (gene 1 and 3) the column has 'moved' to the location shown in Figure 6.10. Column 9 of child B inherited its column characteristics from column 9 of parent 1. The difference, however, is the thickness value, which is 'small' for parent 1 and 'large' for child B.

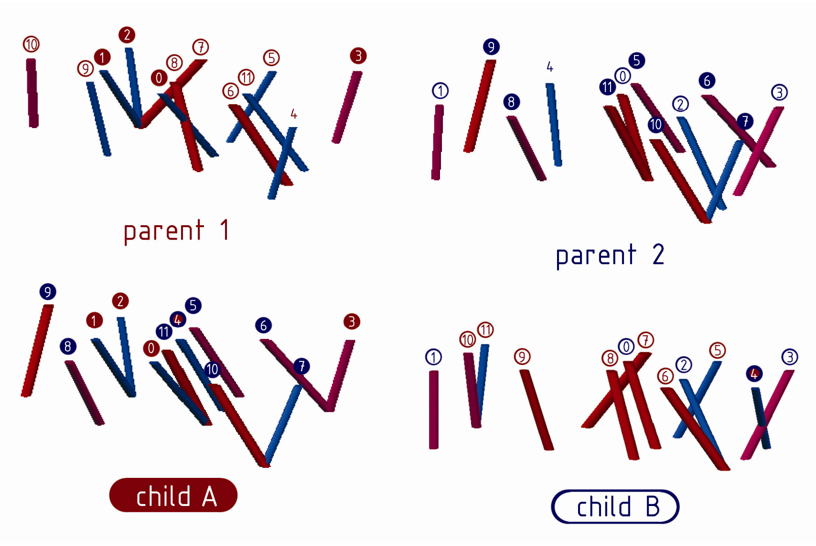


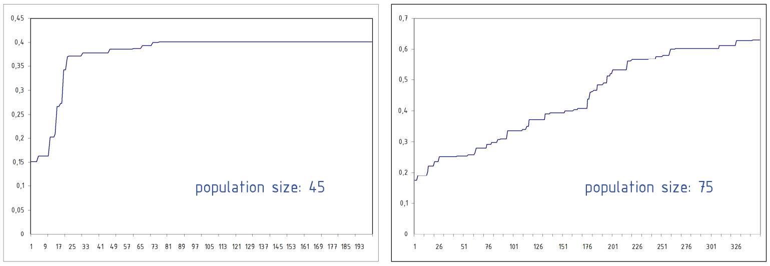
Fig. 6.10 Crossover and mutation of two individuals

6.2.2 optimal values for the crossover and mutation rate

Numerous test runs have to be carried out in order to be able to estimate the optimal values for

- the crossover rate (fixed, but can be variable),
- the mutation rate for the first generation,
- the halving time for the mutation rate [see Section 5.2.5], and
- the population size.

As shown in Figure 6.11, the difference in convergence and ability to work towards an optimal value are considerable, considering the GA parameter values that are used. Because of the enormous variety of possibilities and the fact that every problem has its own set of optimal GA parameter values to come to an optimal solution, in this text further elaboration on the test runs concerning the estimation of optimal GA operator values is refrained from.



crossover rate: 0,50
mutation rate(0): 0,50
halving time: 5 generations

crossover rate: 0,55
mutation rate(0): 0,10
halving time: 45 generations

Fig. 6.11 Improvement of the volume fitness parameter for different operator values

6.3 clarification of the fitness parameters

In order to prevent difficulties with possible divisions by zero, non-fit solutions are valued with a value close to zero e.g. 1.10^{-6} or are given a penalty by multiplying the fitness function with a certain value between zero and one . Fully fit solutions, will have a value of 1 (concerning the parameter under consideration).

stability fitness parameter

The *stability fitness parameter* is the only fitness parameter that has to be complied fully. In other words, a non-fit *stability fitness parameter* value is the result of a structure not capable of transferring any external load, meaning that a solution with a *stability fitness parameter* smaller than one (i.e. 1.10^{-6}) is unstable. The stability of a structure is dependant on the following structural characteristics;

1. the work lines of all columns intersect in one point;
2. the planes between the grid lines described by the work lines of all columns are parallel;
3. less than 6 columns are active.

If one or more of these characteristics is/are applicable to a column structure generated between two grid systems, the structure describes a structural mechanism. As a result, the *stability fitness parameter* value will be close to zero.

bearing capacity fitness parameter

Every column individually is tested for the contribution to the overall *bearing capacity fitness* which places the column in one of the following three categories:

1a. the normal load is larger than zero (tension)
1b. the normal load is zero

> the individual column fitness is close to zero.

2. the normal load exceeds the bearing capacity of the column

> the individual column fitness is given a penalty value proportional with the difference between the normal load and the bearing capacity.

3. the normal load is smaller than zero and does not exceed the bearing capacity

> the individual column fitness is scaled between zero and one proportional to the normal load and the bearing capacity.

The main goal of this fitness parameter is to drive the normal loads of all columns towards the bearing capacity, in order to find a fully stressed structure as an optimal solution.

total column volume fitness parameter

The value of the fitness parameter for the total volume of the columns per individual is dependant on the possible maximum and minimum summed volume. The maximum value for the total column volume is calculated by multiplying the number of columns with their maximum length and maximum cross section area. A similar formula is used to calculate the minimum value for the total column volume, where the minimum length (the construction height) and the minimum cross section area are multiplied with the number of columns.

Based only on the fitness parameter for the total column volume, a higher fitness for a column structure will be driven by the increasing number of (almost) vertical columns with the smallest possible cross section.

total column slanting fitness

Although at least the fittest column structure eventually needs to be evaluated for multiple load cases, good values for the *total column slanting fitness parameter* will result in structures better capable of dealing with multiple load cases. This as a result of spreading the slanting angle of every column, weighted with the column thickness, such that the summed column angles in x- and y-direction are driven towards zero. It needs to be noted, that fitness parameters close to '1' concerning the total slanting of the columns do not automatically ensure good characteristics for dealing with multiple load cases.
In order to give more information on whether a structure is capable of dealing with multiple load cases, without actually subjecting it to a loop of calculations, the variation of column directions and location [see paragraph on *column genes variation fitness parameter*] is of great importance.

intersecting columns fitness parameter

The fitness of a column structure concerning the desire for non-intersecting columns is dependant on two aspects, being the summed volume of every column of the structure individually and the volume of the unioned column structure. The latter can be determined with the AutoCAD-function *Union*, which creates a composite solid by addition. Figure 6.12 shows the difference between the summed and unioned volume of two identical columns.

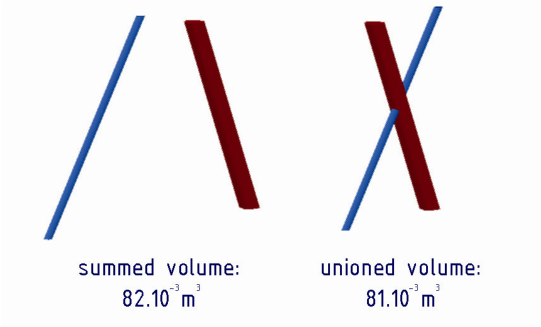


Fig. 6.12. The summed volume of two columns versus the unioned volume of the two columns intersecting

The difference between summed and unioned volume is the intersecting volume, which is the parameter to be minimised in order to maximise the fitness value. This, meaning that, if the summed volume of a column structure is equal to the unioned volume, and the intersecting volume is zero, the *intersecting columns fitness parameter* is valued '1'.

In order to include a minimal distance between the columns, *column profiles* can be utilised within the *intersecting columns fitness parameter*. The same principle of comparing the summed and the unioned volume as describe above can then be applied to the column profiles, which form a larger silhouette around the columns [Figure 6.13].

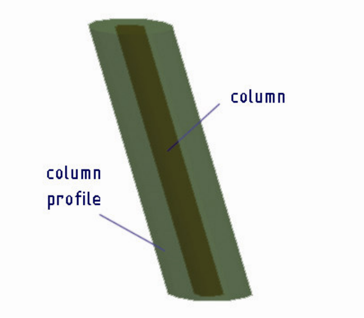


Fig. 6.13. In order to be able to denote a minimal distance between the columns, a column profile can be utilised

As the difference in volume between the summed and unioned structure is relatively small, a bandwidth is adopted in order to prevent that all fitness parameter values are close to '1'. This results in a fitness of '1.10^{-6}' when the quotient between the summed and unioned volume is smaller than e.g. '0,95'. Values larger than '0,95' and smaller than '1' are rewritten to values between '0' and '1'.

Here it must be stated that whether the *intersecting columns fitness parameter* is utilised, the maximum number of possible generations and individuals is reduced considerably, when virtual memory errors can occur. This is due to the fact that drawing the columns and/or the profiles demands a considerable amount of virtual memory of a computer.

column genes variation fitness parameter

The *column genes variation fitness parameter* consists of three independent variation parameters that can be individually weighted within the VBA script;

- variation of column thicknesses
- variation of the angle around the vertical (z) axis
- variation of the column position in relation to the centre of gravity

Main goal of the combined variation fitness parameters is to create structures that appear random to the eye. All three gene variation parameter values are based on different possible column characteristics. The variation of the columns thickness is spread over the three possible column thickness gene values; small, medium, and large. For the indication of the slanting direction of the columns four equally sized quadrants, based on an orthogonal system are denoted [Figure 6.14].

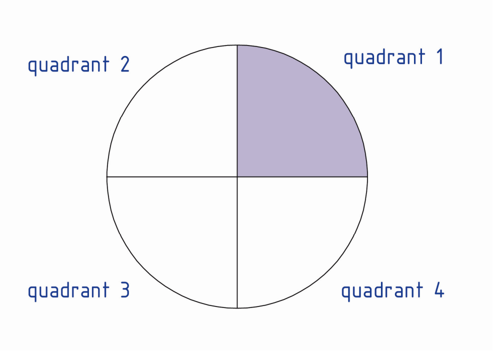


Fig. 6.14. Four different possible quadrant values for the slanting direction of the columns

And finally, the column position in relation to the centre of gravity is spread over four circular areas, increasing in size, indicating the distance to the centre of gravity [Figure 6.15].

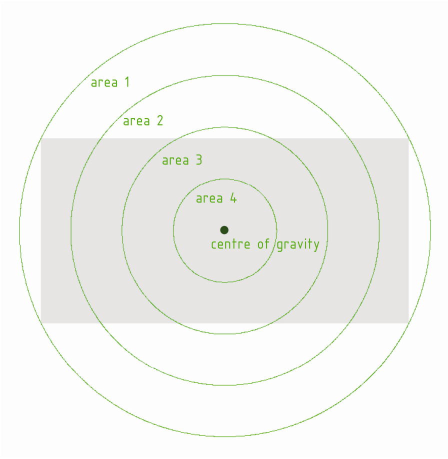


Fig. 6.15. Four different possible values for the distance of the top point of the columns to the centre of gravity

The optimal values for the different gene variation parameters are obtained by an equal distribution of all active columns over the three possible column thickness values and the four possible column slanting direction quadrants for respectively the variation of column thicknesses and the variation of the angle around the vertical axis.
The optimal variation of the distance of the top point of the column to the centre of gravity is accordingly the distribution of the number of active columns per area as given in Figure 6.16, where every area will have half the number of columns in it of the previous class.

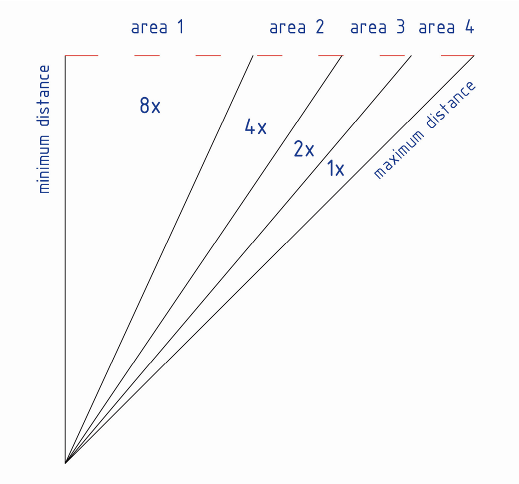


Fig. 6.16. Optimal distribution of the four different possible class values for the distance of the top point of the columns to the centre of gravity

Example

Suppose 15 active columns determine one solution. The maximum distance is 40 m. Class 1 will contain columns at a distance larger than 30 m and smaller than 40 m. Class 2 contains columns at a distance larger than 20 m and smaller than 30 m, and so on. The optimum variation is then that 8 of the 15 active columns are in class 1, 4 columns are in class 2, 2 columns are in class 3, and 1 columns is in class 4.

Now suppose that the number of active columns cannot be divided by 15, for example, the structure consists of 12 active columns. The optimal variation per class is then the integer portion of the outcome of the number of active columns, multiplied by the class factor (i.e. 8, 4, 2, or 1), divided by 15, and adding 0,5. For a number of 12 active columns this results in the following optimal division.

class 1:	*real value*	$12 * \frac{8}{15} + 0,5 = 6,9$
	integer value	6
class 2:	*real value*	$12 * \frac{4}{15} + 0,5 = 3,7$
	integer value	3
class 3:	*real value*	$12 * \frac{8}{15} + 0,5 = 2,1$
	integer value	2
class 4:	*real value*	$12 * \frac{1}{15} + 0,5 = 1,3$
	integer value	1

Besides creating a seemingly random structure, the *column genes variation fitness parameter* also prevents the work lines of all columns of going through one point and prevents the work lines of all columns being parallel.

6.4 concluding remarks

One of the main features of the optimisation design tool is the ability to calculate the normal column forces of the structural transfer zone, based on the DEM method. In comparison with the structural analysis tool Oasys GSA, the results are satisfactorily. One important comment that needs to be made, is that it has not been checked to what extent the model with the rigid superstructure falls short with the real world aspects. That is why, again it is stressed out that this design tool is applicable mainly in the preliminary design stage.

Another shortcoming is that the utilisation of the *intersecting columns fitness parameter* demands too much virtual memory from AutoCAD to work with every desired number of generations, individuals or columns. However, test runs have proven that with a small maximum column length and not to many columns in relation with the size of the transfer zone area, intersection of columns occurs rarely.

7 concluding remarks and recommendations

This chapter concludes book one of the Master's thesis 'optimisation of structural transfer zones in multi-use buildings'. Here, conclusions are given in two distinct sub sections. The first deals with conclusions on the utilisation of optimisation techniques in the (structural) design process. The second sub section deals with conclusions on the development of the design tool.

general conclusions

Traditionally, structural optimisation is used as a tool for reducing the weight of structures and structural components, with at this moment of time the accent on the latter one. The purpose of many approaches, is to produce a design in which each structural element sustains a stress level as close as possible to the allowable stress of the material under given loads and support conditions, with the optimal target of a fully stressed design. This aim for reducing the structural weight based on structural optimisation is already extensively used in aerospace engineering, where reducing the weight is of the utmost importance.

However, since every design problem requires different parameters and boundary conditions, it is impracticable to create general tools applicable for a large set of problems without the use of libraries. This implies that for every new problem a new tool has to be designed, programmed, or scripted from scratch. This may sound like an approach including a lot of redundant work, but the main advantage is that the structural engineering, possibly together with a computer scientist, can work towards a comprehensible and straightforward script for the design tool. Scripting for a specific design problem also has the advantage that the tool not needs to be 100% fool proof, since the intended user (in most cases the designer him- or herself) may be considered 'not a fool'.

To conclude the general conclusions, an interesting question could be, whether with such tools, a structure is still designed or if it is computed or calculated, and if the latter is true, is this a problem? It is claimed here, that however the influence of calculation versus the creative aspect of design is relatively high, the transfer zones in this Master's thesis are still designed. This, as the structured aspects of generating the structure based on GA's are complemented with the creative process of designing the model environment and parameters.

conclusions on the utilisation of optimisation techniques

usability and applicability

- The potential of structural optimisation techniques has not been realised by the building engineering industry. Actually, there is a sever imbalance between the enormous amount of literature on the subject of structural optimisation and the lack of applications to practical design problems. However, the growing utilisation is a fact, and a result of the increase of technological possibilities and the development of efficient techniques for several practical applications.
- Although, the past decades have produced several original structural optimisation methods, the future will focus on only a few reliable and robust techniques, of which genetic algorithms is one.
- The specific needed knowledge to work with structural optimisation – integration of CAD tools for geometric modelling, general analysis methods, like the Finite Element Method, methods of design analysis with mathematical and mechanical programming, and artificial intelligence techniques – is not something which can be easily asked from one person.
- There is always the threat of users relying fully on the outcome of the optimisation tool. That is why, it is concluded here that small, relatively easy to comprehend design tools, maybe even developed by the user self have more advantages for grasping the essence of a design over a comprehensive, extensive tool.

developer and user

- The future of structural design tools similar to the one describe in the Master's thesis lies not in the generation of (templates of) general tools by experts, but in the creation of awareness and education of structural and building engineers in computational design, and in emphasising the collaboration with computer scientists.
- The building practice will then benefit, mainly in the concept and preliminary stage of the design process from a quick exploration of possibilities, without having to use a 'black box' of which the results can not be derived.
- As the wish/demand for special structures is growing, the number of projects in which design tools can and will be utilise will increase accordingly.

conclusions on the development of the design tool

aspects of the design tool

- As structural optimisation techniques in building engineering are utilised mainly for special structures at this moment (e.g. free form structures), design tools are usually programmed for a specific design problem (consider the Folded Roof Project, Section 5.4 of book two).

- Although the onset of the design tool described in the Master's thesis is applicable for many structural design processes including a column allocation problem, the presented script linked to AutoCAD itself is another example of a rather specific design tool. The presented design tool amongst others (i) can only be used for vertical line elements (i.e. no walls), (ii) is limited to just one structural layer, and (iii) contains specifics parameters, such as the discrete location possibilities, and column sections.
- As scripting and programming are often interlinked, it is stated here that the VBA implementation presented in this Master's thesis is a scripted tool – not a program – which can be utilised within the AutoCAD environment.
- It can be concluded that with the six objectives, optimisation of the fitness function is achievable to a very large extent. Mainly because the problem adopted in this Master's thesis can be well 'tackled' by an optimisation methods such as genetic algorithms, and that the optimisation process as a result of running the VBA script can be well implemented in the preliminary design stage.
- It can also be concluded, that for a complete benefit of the GAs a computer scientist and a structural engineer should combine their knowledge in making a scientific computer model for building engineering purposes. First of all, because this collaboration will often lead to new insights in both fields. Secondly, both academics can display their own specific knowledge to a problem, rather than needing one building engineer with extended computing knowledge.

recommendations

- The convergence to the optimal solution might improve by choosing other types of GA operators. It might be that multi-point crossover instead of single point crossover, in-order mutate instead of random mutate, and best-individual replacement instead of parent-replacement will result in better optimisation systems.
- Currently, it is not possible to deal with multiple load cases, however, implementing a loop to make it possible is probably not that hard to script. Another applicable feature could be using a manual stop during the run to fix desired columns and thus to ensure that these columns will remain in every generation.
- It can also be recommended to involve the optimisation of the principle beams on the grid system in the optimisation tool, and to enlighten the transfer of the loads from the superstructure, which are introduced in the columns. Another important structural aspect, not implemented in this Master's thesis, is the structural integrity of the transfer zone. On the other hand, such aspects can be invoked after the optimisation process.

- To answer to the free form desire currently popular in the building industry, the tool could be adapted such that curved 'grid lines' are also able to offer a framework for the structural transfer zone generated with the VBA script. And although the number of objectives can not be too large, some (structural) aspects could be adopted as well, such as the constructability, construction time, the costst, the adaptivity and a value for the functionality of the transfer zone.
- This research does not go into detail on the question whether grids are still the principal for designing a (multi-use) building, but with a topic on an optimisation process to connect different structural grids, this question might need an answer in the successor of this Master's thesis.

assumptions

- Applied on the design tool for structural optimisation of a transfer zone, it may be assumed that the user will not introduce random or impossible values for the different parameters (such as a negative value for the number of columns), or at least will recognise deficiencies in the outcome after validation the results.

references

books and reports

[1] Bartlett, G., *Practical handbook of genetic algorithms – Applications, volume 1*. Boca Raton: CRC Press Inc., ed. L. Chambers 1995

[2] Barron, D.W., *The world of scripting languages*. Southampton: John Wiley & Sons, Inc. 2000

[3] Burns, S.A., *Recent advances in optimal structural design*. Reston: American Society of Civil Engineers, 2002

[4] Coenders, J.L., *Master's thesis – Form finding and structural optimization, research and development of a new concept*. Delft: Delft University of Technology, Structural Design Lab, 2004

[5] Engelbrecht, A.P., *Computational intelligence – an introduction*. West Sussex: John Wiley & Sons Ltd., 2002

[6] Escher, M.C., *M.C. Escher – Grafiek en tekeningen*. Köln: Taschen GmbH, 2001

[7] Groot, S.H. de, *"De Componist" – ontwerp voor een woongebouw te Amstelveen*. Delft: Delft University of Technology, 2004

[8] Hartsuijker, C., *Toegepaste mechanica, Deel 2 Spanningen, vervormingen en verplaatsingen*. Schoonhoven: Academic Service, 2000

[9] Hartsuijker, C., and C.F. Vrijman. *Mechanica van constructies 2a – statisch onbepaalde constructies*. Delft: Delft University of Technology, 2000

[10] Liang, Q.Q., *Performance-based optimization of structures – Theory and applications*. Oxon: Spon Press, 2005

[11] Lutke Schipholt, R., *Overspannend staal – basisboek*. Third edition. Rotterdam: Staalbouwkundig genootschap, 1996

[12] Mitchell, M., *An introduction to genetic algorithms*. Cambridge: The MIT Press, 1996

[13] Nederlandse norm NEN 2443 (nl), *Parkeren en stallen van personenauto's op terreinen en in garages*, 2000

[14] Neufert, E. *Architects' data*. Third edition. London: Blackwell Science. 2004

[15] Pfeifer, R., and C. Scheier., *Understanding intelligence*. Cambridge: The MIT Press, 1999

[16] Quispel, E., *Onderzoeksrapport – functiestapeling boven ondergrondse parkeergarages*. Delft: Delft University of Technology, 2004

[17] Raven, W.J.., *Vuistregels voor het bepalen van afmetingen van vloeren, balken en kolommen in hout, staal en beton*. Delft: Delft University of Technology, 2003

[18] Scheers, P.M.C., *Structuur*. Delft: Delft University of Technology, 2001

[19] Scheurer. F., *The Groningen Twister – an experiment in applied generative design*. Milan: In Generative Art 2003. Proceedings of the 4th International Conference, ed. C. Soddu, 2003

[20] Schodek, D.L., *Structures* Fifth edition. New Jersey: Pearson Education, 2004

[21] Strasky, J., *Architecture of bridges as developed from the structural solution*. FIP International Congress on Prestressed Concrete. Washington DC, 1994

[22] Vamberský, J.N.J.A., et al., *Designing and understanding precast concrete structures in buildings*. Delft: Delft University of Technology, 2002

[23] Wagemans, L.A.G., *Infomap algemene constructieleer*. Delft: Delft University of Technology, 2000

internet

[24] Chaitin, G.J., *Randomness and mathematical proof*. September 2006.
http://www.cs.auckland.ac.nz/CDMTCS/chaitin/sciamer.html

[25] ONX architecten bna, *ONX architecten*. November 2005.
http://www.onx-architecten.nl

[26] Random.org, *True random number service*. September 2006.
www.random.org

[27] Massachusetts Institute of Technology, *GAlib: FAQ*. June 2006.
http://lancet.mit.edu/ga/FAQ.html

appendix A - Visual Basic for Application as a language[38]

Many programming languages can be used to script the genetic algorithm code for the particular problem of allocating columns in transfer zones. Among these, Visual Basic for Applications (VBA), Processing – a Java-based open source programming language and environment, including a library for particle spring systems [Appendix E, book two] -, and GenerativeComponents are accessible. Considering the latter one, the director of research of Bentley Systems, inc., Robert Aish, PhD, states that GenerativeComponents is 'a parametric and associative design system for architecture, building engineering and digital fabrication. It makes use of an advanced parametric engine to unify all aspects of design and allows designers to build geometric models based on components and inter-component relationships'. The question of course is, whether a single system can deal with the unification of all aspects in (structural) design. And just like with Processing, this language is only presented in its beta-version.

The language best known by the author is VBA by Microsoft, which is an event driven programming language. Visual Basic for Applications was designed to be usable by all programmers, whether novice or expert. The language is designed to make it easy to create simple GUI applications, but also has the flexibility to develop fairly complex applications as well. Programming in VBA is a combination of visually arranging components on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. By writing the script in VBA in the AutoCAD-environment, clear graphical output is fairly easy to produce.

---

[38] Barron

Master’s thesis

9,8 **appendices**

appendix B - flowchart of the VBA script of the design tool