ANT SYSTEM BASED STRUCTURAL DESIGN OF A ROOF IN ULTRA-HIGH PERFORMANCE CONCRETE
MSc Thesis

M.G. Flint
June 2008

Delft University of Technology
Faculty of Civil Engineering
Structural Design Lab

In collaboration with Corsmit Consulting Engineers

Supervisors:
Prof.dipl.-ing. J.N.J.A. Vamberský
Prof.dr.ir. J.G. Rots
Ir. J.L. Coenders
Dr.-ing. S. Grünewald
Dr.ir. M.A.N. Hendriks
Ir. J.W. Welleman

TU Delft
Delft University of Technology

structural design lab

CORSMIT
RAADGEVEND
INGENIEURSBUREAU
BV

TABLE OF CONTENTS

Go to the ant, thou sluggard; consider her ways, and be wise:
- Book of Proverbs 6:6

PREFACE

This report comprises the results of my Master's thesis. The thesis focuses on making advances in computational optimisation methods for structural topology design, combining this with a new material and applying it to a real-life case. Three different fields that are linked to structural design are connected in this way.

This report is built up out of eight chapters. Chapter 1 is an introduction to the thesis. It gives the context, outlines the challenges and defines the objective. Chapter 2, 3 and 4 focus on the relevant literature that has been studied. They describe background information of optimisation in structural design, the ant system and ultra-high performance concrete, respectively. Chapter 5 describes the optimisation algorithm that has been developed and Chapter 6 elaborates on the results of the validation and the different sensitivity analyses. The application of this algorithm to a case results in the design that is discussed in Chapter 7. Conclusions and recommendations are listed in Chapter 8.

Starting in October 2007, the thesis has taken eight months to complete. With this report a very intensive period comes to an end. It has not always been easy to stay focused on such a long term, individual and in-depth project. Completing this thesis and my studies indeed would not have been possible without the help of some people I would like to thank at this point.

First of all, I would like to thank my supervisors prof.dipl.-ing. J.N.J.A. Vamberský, prof.dr.ir. J.G. Rots, ir. J.L. Coenders, dr.ir. S. Grünewald, ir. J.W. Welleman and dr.ir. M.A.N. Hendriks. Their ideas have been both helpful and very inspiring. I would like to thank Imke, Anke and Maryn for reviewing the report and suggesting improvements and Jan Jouke for his help on the lay-out. I would like to thank my colleagues at Corsmit for their interest in my project, their expertise and critical opinion which helped me putting my ideas on computational design in perspective. Especially Jeroen Luttmer, Harm Kraaijenbrink and Jaap Boon have been of great help. Thanks to my fellow students at room 0.72 for the good atmosphere and the sharing of knowledge and ideas. I would like to thank my family and my parents in particular for their support during my studies. They have been very enthusiastic about everything I have done since I arrived in Delft seven years ago and have always shown their confidence in me and my decisions. Lastly I would like to thank my friends who made my time as a student unforgettable. Special thanks to my *jaarclub*, which proved a great group of friends. It is amazing how we have developed over the years to a great support for each other.

Bedankt!

Martin Flint
Delft, June 2008

SUMMARY

Objective

The recently developed material ultra-high performance concrete (UHPC), introduced by Richard and Cheyrezy [50], has compressive strengths of over 150 MPa and a ductile behaviour. It has a higher stiffness and superior durability characteristics in comparison with ordinary concrete [1]. The opportunity emerges to find new optimal structural topologies for this material [58]. Developing topologies for unexplored materials can be a process of trial-and-error or logical deduction, such as happened in the past for many new materials. A third approach is the use of computational optimisation algorithms. A new material provides opportunities for innovative solutions, and this approach could provide out-of-the-box ideas about what topologies could be optimal for UHPC.

The computational optimisation algorithm called the ant system is able to solve combinatorial optimisation problems [27]. The ant system and its extensions are nowadays all referred to with the term ant colony optimisation (ACO). Their optimisation process is based on the foraging behaviour of ants. The artificial agents all follow simple probabilistic rules and communicate indirectly through laying *pheromone* [8]. ACO is not widely explored for structural design cases. Given the nature of the algorithm, ACO can be applied to topology optimisation if a right formulation and representation for the problem is found.

In the city of Apeldoorn a new sports centre is under construction. The centre is called Omnisport and consists of several halls. One of the halls contains a cycling and athletics track. The roof of this hall spans an elliptic area of about 120 by 100 metres. The load-bearing structure of this roof consists of steel trusses spanning the shortest distance. Several different structural design options for this roof are possible.

This thesis focuses on the creation and evaluation of a structural topology design algorithm, based on the concept of the ant system, that designs a roof in ultra-high performance concrete. The algorithm is applied to the design case of the Omnisport sports hall roof.

Algorithm

A new application for ant colony optimisation is developed in this thesis. Structures are mapped to a binary search space, which forms the link between the structure, ant colony optimisation and the finite element package DIANA. Elements in a meshed area are randomly assigned material or left a void by the agents. According to the performance of the structure that an agent generates in this way, the probability that material is assigned to each of the elements in the mesh is altered for the next iteration. Elements that together form a well-performing structure are more likely to be chosen again.

Because the process is based on a performance that can be any function, the approach has a wide range of possible applications and provides the opportunity to optimise towards other criteria than structural considerations only. In the case considered in the thesis, the UHPC structure is optimised towards a combination of minimum volume, minimum mould surface, minimum pre-stress and availability of holes for ducts and walking bridges. The algorithm performs multi-objective optimisation resulting in a structural topology. The concept can also be applied to three-dimensional problems.

The algorithm has been validated on least weight problems on which it gave a priori expected results. Manual checks have been performed to confirm the correctness of the calculations performed within the algorithm. The influence of the parameters that are set at the start of a run has been researched. As a result, a parameter set that yields satisfactory runs is determined as well as guidelines for the possible adjustment of this set. The influence of the performance function has been researched and the robustness of the algorithm is demonstrated by applying it to several different cases and boundary conditions. Resulting structures have a possible optimal topology, that needs to be carefully interpreted, analysed and developed by an engineer or designer.

Design

The properties of UHPC have been implemented in the algorithm, and the algorithm has been applied to the case of the Omnisport hall roof. Based on the results, a preliminary design for the load-bearing structure of the roof has been made. A load-bearing structure in this material is found to be possible. The design consists of truss-like supporting elements for the roof. A connection strategy is proposed; prestressing strands in the members are anchored on stiff steel

end plates that are attached to both ends of a member. The different steel plates that come together in a node are welded to a steel plate that has a perpendicular orientation to the end plates. Member sizes and necessary pre-stress are indicatively determined. Rough cost estimates show that the design is not necessarily more expensive than the current design in steel. The resulting design is preliminary and should be checked and refined.

Concluding remarks

A new application for the ant colony optimisation meta-heiristic has been developed for multi-objective structural topology optimisation. The applicability of the algorithm to structural design cases is demonstrated through a case. The method has some drawbacks. Runs can stagnate prematurely and the optimisation runs are time-expensive. The runs result in a topological design. Resulting structures need to be carefully analysed and developed by an engineer or designer. Refinement of the tool is possible and advisable.

The thesis shows that the superior properties of ultra-high performance concrete in comparison with ordinary concrete lead to the situation where different structural topologies like trusses become efficient.

CHAPTER 1 INTRODUCTION

1.1 Context

The construction industry is a field of work where several scientific areas come together. Materials, construction techniques, structural mechanics and structural design techniques are some of them. Because they are linked, developments on any of the fields influence each other.

In the field of materials science, *ultra-high performance concrete* (UHPC) is rapidly developing. The compressive stresses that can be allowed in this material are a multiple of those that can be allowed in conventional concrete. Therefore opportunities arise to find new optimal structural topologies for this material that are not necessarily based on the knowledge and experience developed with conventional concrete [58].

At the same time one can see developments in the way structures are designed. Several computational algorithms simulate the natural processes of growth and evolution and lead to innovative and efficient designs. Newly developed optimisation techniques like *evolutionary structural optimisation* (ESO) [19] and the concepts of *genetic algorithms* (GA) [12] and *simulated annealing* (SA) [39] are occasionally applied to structural design cases. Another algorithm is the *ant system*, introduced by Dorigo et al. in 1996 [27]. Nowadays, ant system based optimisation algorithms are referred to with the term *ant colony optimisation* (ACO) [8]. Though the concept seems promising, it has only had limited applications in structural design.

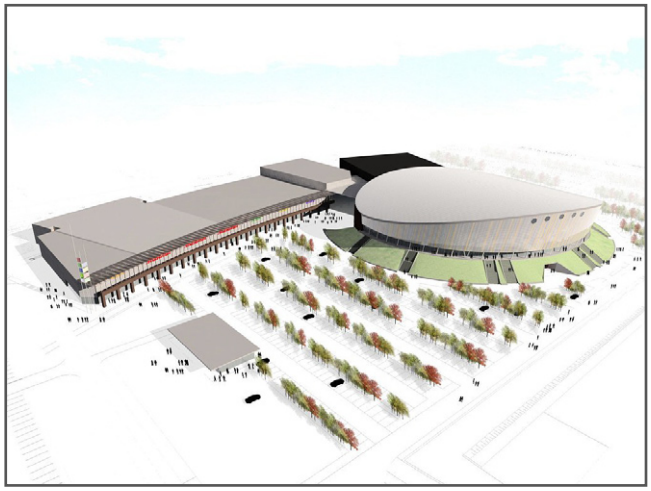


Figure 1.1: Omnisport sports centre. Artist impression of the current design.

This situation leads to a unique opportunity. By combining a new material and a new way of developing structures, innovative solutions can be found. Without being prejudged by experience with other materials or structures, an algorithm can be used as a tool that proposes optimised solutions to the engineer or architect.

1.2 Project outline

1.2.1 Ultra-high performance concrete

Concrete quality is measured in its characteristic compressive strength. Whilst conventional concrete has a quality of about C40 (characteristic strength of 40 MPa), ultra-high performance concrete can have qualities higher than C200 which means its strength is five times as high [13]. The extreme stresses this material can deal with, together with its higher stiffness and better durability, lead to new opportunities for structural designers. The new material asks for new, out-of-the-box structural concepts [58]. The following assumption is made.

UHPC with its improved properties provides the opportunity to optimise structures accordingly.

Because the range of possibilities for the improvement of the application of UHPC is vastly unexplored terrain, research in this field is required. The fact that only preliminary codes are available asks for a thorough understanding of the material behaviour by the designer. These reasons make applying UHPC in this thesis very interesting.

1.2.2 Ant system based optimisation

In the changing world of structural design, iterative optimisation algorithms based on concepts found in nature gain attention. Several evolutionary concepts have been applied

and have found to be valuable as an optimisation method in structural design. The ant system, introduced by Dorigo et al. in 1996 [27] is promising and uniquely exploits an indirect form of memory of previous performance [8]. It is mainly used in logistics engineering up so far. Discrete steel truss optimisation is the most widely explored of the few applications of ant colony optimisation in structural engineering. It is probable that wider applications are possible and of value in the structural designing industry. This leads to the following observation.

Ant colony optimisation is not widely used as an optimisation algorithm in structural design, even though it is a promising concept.

It is chosen to develop an algorithm and use it for designing the UHPC structure. This approach can provide structural concepts that are not limited to knowledge and experience in the way they would be when designed by an engineer. Novel solutions are asked for when using this material. The ant system concept is chosen because it can take into account more performance criteria than for example ESO [19] and it uses the generated knowledge differently than for example GA [12] or SA [39]. The concept of the ant system is not widely applied to structural topology optimisation. Topology optimisation based on the ant system is innovative and exploring the possibilities and drawbacks for this is interesting from a research point of view.

1.2.3 Omnisport building

In the city of Apeldoorn, located in the east of the Netherlands, a multi-functional sports centre is under construction. From 2008 onwards, this will be the first accommodation in the Netherlands that meets the Olympic standards for indoor athletics and track cycling. The building consists of a volleyball hall, a cycling and athletics hall and a central area connecting the two buildings. The roof of the cycling and athletics hall spans an elliptical area with a width of 100 metres and a length of 120 metres. In the design, steel trusses span the shortest distance and girders are placed in between the trusses supporting the roof. This is a well-proven economical option. For this kind of spans, however, several solutions are possible. Modern optimisation methods in combination with the use of new materials can lead to different solutions. Figure 1.1 shows a render of the project, and technical drawings can be found in Appendix A. The following hypothesis is stated.

Several structural design options for the Omnisport cycling hall different than the chosen one are possible.

The Omnisport roof is chosen as a case in this thesis for four reasons. First of all, it is interesting to apply the design algorithm to a case and take it out of the theoretical context because it pushes the designer to think about practical problems. Secondly, the large span is interesting; a special structure can add value to the experience in the building. Thirdly the newly developed solution can be compared with the current design. Finally, the structural designers gave access to the drawings of the case and are willing to share information and reasoning behind the decisions they made.

1.3 Objective

Three issues have been observed concerning a material, a design method and an application. This Master's thesis will combine the three observations. A design algorithm will be developed and applied to the Omnisport case. The algorithm will be based on the ant system, a concept that might provide an innovative way of coming up with novel topological designs. This approach will be used in combination with UHPC, a material that asks for new approaches and designs. The challenge of this Master's thesis sounds:

To create and evaluate a structural design algorithm, based on the concept of the ant system, that provides conceptual ideas for the structural design of a roof for the Omnisport building in Amersfoort in ultra-high performance concrete.

CHAPTER 2 OPTIMISATION IN STRUCTURAL DESIGN

Optima are subjective; what does one define as an optimum? In building design, one can optimise to all sorts of things: looks, practicality, safety, thermal insulation, costs, weight, sustainability and labour are only a few of them. A logical but very subjective step when starting an optimisation is to determine how performance is valued. If one has valued performance, one can optimise that performance in an objective manner. This is where mathematics comes into play. Having valued performance one can seek for the best performance by altering certain parameters. Complexity is now added by the many parameters in building design. Having many parameters, a multidimensional design space is obtained. Yet a building design project has an almost unlimited number of variables.

Optima are never really optima in every way, but finding a certain optimum is a good guidance in designing [38]. Subjective judgement of good and bad and valuing priorities is a necessary step that will never be uniformly agreed on. Keeping this in mind enables one to put the found optimum in perspective.

This chapter is by no means a comprehensive overview of optimisation in structural design. It will merely outline the background of structural optimisation, and shortly describe the theoretic and biomimetic concepts that are most important for this thesis. The concepts described in this chapter are either a benchmark or a possible source of inspiration for the algorithm that will be developed in this thesis.

2.1 Optimal shapes

In this paragraph traditional optimisation methods are discussed. Shells and membranes can be designed using physical form finding. Trusses have an optimal shape when designed like Michell did over a hundred years ago [15]. However brilliant the methods may be, the form finding methods introduced here are all optimal with respect to weight and one kind of loading. Hence, they are partially optimised and not very robust.

2.1.1 Shells and membranes

Being structures of pure normal stress, shells and membranes are easily form found shapes. Several physical methods for finding optimal shapes have been developed over the years, of which the most important are listed here.

Hanging chain models

It was Anthony Gaudí who made the hanging physical models a world-famous form finding technique [5]. A hanging chain shapes itself to a configuration in which only tensile stresses occur. When the same shape is turned upside down, the configuration is found in which only compressive stresses occur in the structure. Hanging models are a great way to generate optimal shapes when self weight is the structure's dominant design load and when materials are used that are especially efficient when loaded in pure compression, like

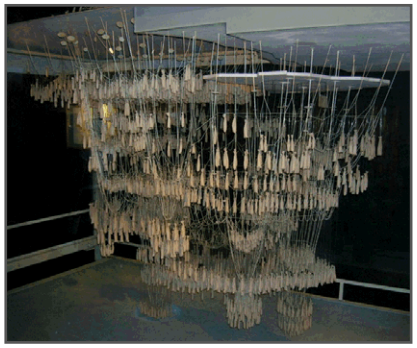


Figure 2.1: Hanging model of Gaudí's Sagrada Familia.

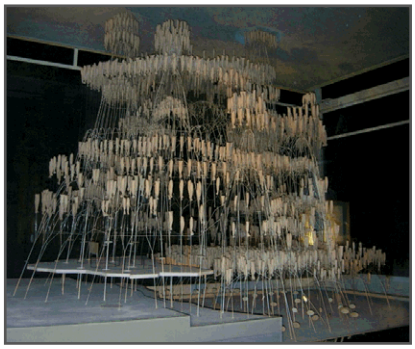


Figure 2.2: Flipped counterpart of Gaudí's hanging model.

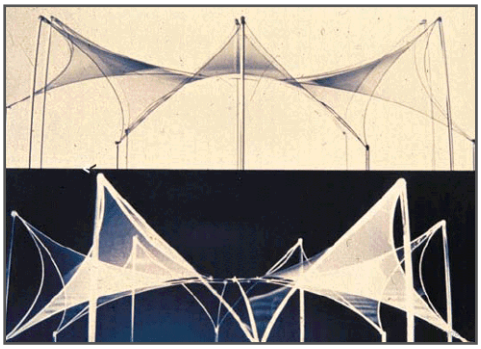


Figure 2.3: Soap film model of a membrane structure. Picture from Lee [41].

concrete.

By modelling loads with small bags of sand, Gaudí designed his Sagrada Familia, Guell crypt and several other buildings using this simple concept.

Soap film models

Soap films minimise their surface energy [5]. This results in minimal surfaces between preset closed boundaries. Interestingly, in a two dimensional plane soap bubbles always come together with three surfaces in one node. The angle between the boundaries of the surfaces is always 120 degrees [5].

Soap film modelling is especially useful when designing pneumatic or tensile structures.

Wet cloth models

Wet cloth models are conceptually comparable to hanging chain models. This form finding technique is especially suitable for shells. The membranes are in pure tension when hanging, while the flipped counterpart is the ideally shaped shell for compression. Heinz Isler is known for designing his very thin shells in this manner [5].

Computational models

Several computational packages simulate physical models quick and easy. The force density method and dynamic relaxation are techniques to simulate the shapes of cable-nets and membrane structures [5]. Whereas the force density method is specifically developed for these purposes, dynamic relaxation is a more general approach and can also be applied for other problems. A famous example of a structure whose geometry is designed using dynamic relaxation is the roof of the British Museum [5]. Computational methods for finding minimal surfaces create digital soap film models.

2.1.2 Michell structures

In 1904, Michell derived a theory for designing structures with a minimum volume of material [15]. His theory is applicable to two-dimensional trusses and consists of truss members following the directions of the principal stresses. Principal stresses are perpendicular to each other by definition, which is why the members of Michell trusses always intersect at straight angles [15].

The optimal structures found by Michell are theoretically sound but very academic in nature. The structures are optimised for one load only, and do not take into account self weight. Practical applications therefore are difficult, but Michell structures provide a good benchmark in optimisation because their shape is proven to be the most efficient.

2.2 Mathematical optimisation

Mathematical optimisation can be defined as the process of finding a maximum or minimum value of an object-function within a given set of boundaries and constraints. Mathematical optimisation routines can be applied more generally than optimal shape techniques. The definition of mathematical optimisation is also the basis for many modern computational optimisation algorithms. Here only a few mathematical optimisation methods that are relevant for this thesis are described.

2.2.1 Lagrange multipliers

The mathematical optimisation theory introduced by the Italian Joseph Louis Lagrange in the 18th century considers multidimensional constrained optimisation. The principal idea is that the minimum or maximum value of a function $f(x,y)$, subject to the condition $g(x,y)=c$ which occurs in point $X=(x',y')$, ∇f and ∇g must have the same direction at X [5, 72, 73].

In other words, if:

$$L(x,y,\lambda) = f(x,y) + \lambda(g(x,y) - c)$$

Then an optimal point is found when:

$$\nabla_{x,y,\lambda} L(x,y,\lambda) = 0$$

2.2.2 Monte Carlo

Probably the simplest concept in mathematical optimisation, the Monte Carlo method compares randomly generated solutions.

Because the solutions generated are random, the Monte Carlo method is not sensitive to getting stuck in local optima. Since the search process is not guided in any way the method is not very efficient. Many calculations need to be done before an optimum is found. However, when a problem has a high number of dimensions, the Monte Carlo method becomes more and more efficient compared to other methods.

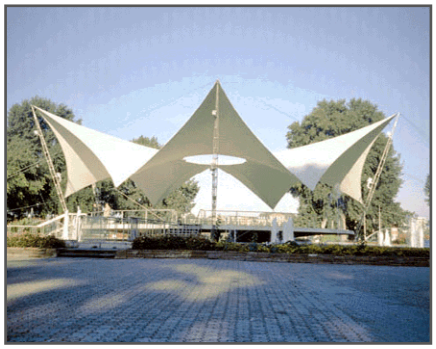


Figure 2.4: Built counterpart of the soap film model. Picture from Lee [41].

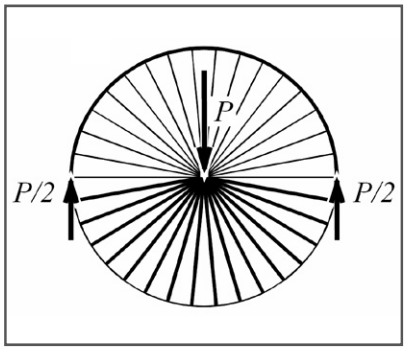


Figure 2.5: Michell structure for a truss on two supports. Picture from Kepler [38].

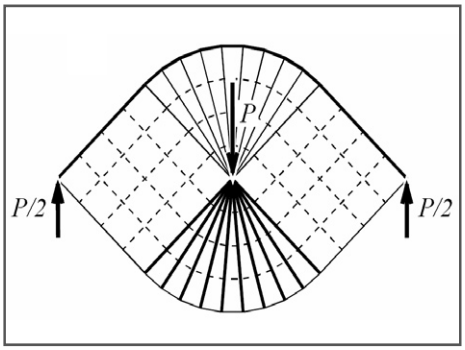


Figure 2.6: Michell structure for a truss on two supports with grid extension. From Kepler [38].

2.2.3 Differential evolution

In the concept of *differential evolution* (DE) three trial solutions are chosen out of a population of trial solutions. Together they determine a location in the search domain where a solution is generated in a next iterative step. By using the data of two trial solutions to adjust a third, the domain is searched in an guided way [9].

The picture in Figure 2.8 shows how DE generates new solutions by adding the weighted difference vector between two population members to the third member. The example is two-dimensional, but the concept also applies to multidimensional problems. The following function shows how a trial location is chosen.

$$X_U = X_{r3} + F(X_{r1} - X_{r2})$$

The weight factor F, $0<F<1$, that weighs the difference between the two members is a control parameter in differential evolution; using a small weight factor guarantees the search domain to be searched thoroughly and lowers the risk of getting stuck in local optima, whereas a large weight factor guarantees finding a solution quickly, but brings along the risk over overshooting the global optimum.

An interesting feature of differential evolution is the way the mutation vector grows smaller throughout the process. Because the evaluated points will get closer to each other when a solution is approximated, the mutation vector automatically gets smaller. This makes sure that the domain will get searched thoroughly around an optimum [9].

As a last note the similarity to the genetic algorithm is brought to attention. Where the genetic algorithm uses two parent solutions to determine a new one, differential evolution uses three. DE can therefore be seen as an extension to genetic algorithms and is often stated to be a biomimetic optimisation concept. However, because there is no truly similar optimisation process found in nature, DE is here argued to be a mathematical rather than a biomimetic process.

2.3 Biomimetic optimisation

After Charles Darwin discovered evolution in nature, awareness has grown that the interesting phenomena found in nature all follow from simple rules. Their complexity results from an evolutionary process. This awareness is now used in optimisation. A new approach towards structural optimisation is gaining popularity; mimicking the formation of natural phenomena, a part of *biomimetics*.

Biomimetic optimisation copies the simple rules found in nature. Optima are found by letting structures evolve over several generations, or iterations. These concepts are especially valuable when used in combination with computers. They can do many iterations in a relatively small amount of time. The principle of analysing, adjusting and analysing again is the basis of every biomimetic algorithm.

2.3.1 Simulated annealing

Simulated annealing (SA) is a biomimetic guided random search algorithm. It is based on the procedure used to make glass as strong as possible. In this procedure glass is heated, so that the atoms can move freely. Cooling down the glass slowly, the atoms find their most stable state, or *lowest energy state*. During the process jumps to higher energy states are occasionally observed. The atom orientation *climbs out* of a local optimum in this way. This process in known as annealing [39].

Annealing is simulated creating a way to guide the Monte Carlo method. First, the optimisation space is explored via the Monte Carlo method. Accepted solutions are stored and the area around them is searched in a next step. A solution is automatically accepted when it is better than the best one found before. If the solution is not better, there is still a chance that it is accepted. The acceptance criterion is important because it determines the parts of the search space that are explored in the optimisation process. The acceptance criterion is represented in the following formula.

$$P_T\{\text{accept } j\} = \begin{cases} 1 & \text{if } E_j \le E_i \\ \exp\left(\dfrac{E_i - E_j}{k_B T}\right) & \text{if } E_j > E_i \end{cases}$$

Where k_B is the Boltzmann parameter that determines the likeliness of accepting non-superior solutions.

Because every solution has a chance of being accepted it is possible to find a better global solution even after the process has been stuck in a local optimum [39].

Analogies with annealing are the following:

- The value found in a location in the optimisation space is analogous to the energy level of the physical system.
- An accepted non-optimal value is equivalent to an occasional jump to a higher energy level of the physical

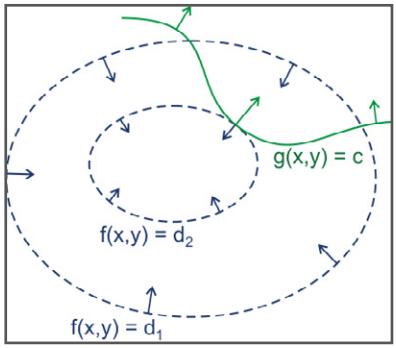


Figure 2.7: Visualisation of Lagrange multiplier. Picture from Wikipedia [72].

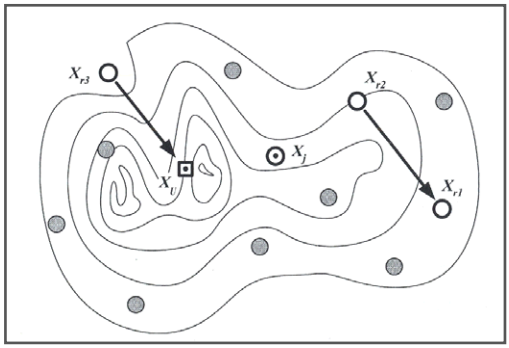


Figure 2.8: Visualisation of differential evolution optimisation. Picture from Feoktistov [9].

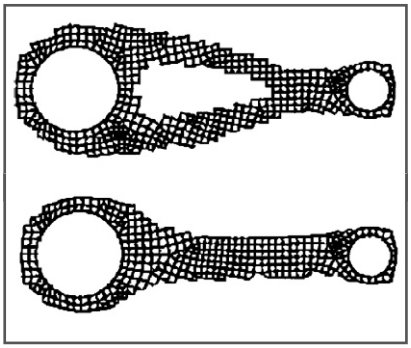


Figure 2.9: SA on a torque arm with different maximum stresses. Pictures from Shim and Manoochehri [54].

system.
- The global optimum corresponds to the lowest energy state of the physical system.
- An iteration in the process is an analogy to a decrease in temperature of the physical system.

2.3.2 Genetic algorithms

Darwin's theory about the survival of the fittest in a population is the concept behind *genetic algorithms* (GA). Genetic algorithms use two separate spaces; a search space and a solution space. The search space is full of coded solutions, and the solution space contains the actual solutions. A mapping process links the two spaces. This is just like one can see in nature; we all are persons, but we have codes in the form of genes that determine our properties [2].

The coded solutions are combined with each other to create new solutions. The new solutions therefore contain some of the genes of one and some of the genes of the other solutions. This process is called *crossover*. The solutions are tested and they are given a value. This value, or *fitness*, determines how likely it is that this particular solution will be used by the creation of new solutions. A higher fitness of a solution means more new solutions are generated with its properties. Occasionally, solutions are copied *(cloned)* into the new generation of solutions. Also, some solutions are not matched with others but just slightly altered themselves *(mutation)* [2, 12].

Analogies with the evolution of a species are:
- Solutions and codes are separated just like genes are not the actual creatures themselves.
- Solutions are combined, just like children have a few properties of each of their parents.

By assigning fitness values to the solution it is possible to pick the best solutions out of a generation and use them to create a new generation. This way the solutions converge towards an optimum. In Figure 2.12 a ground structure and the optimised counterpart yielded by GA optimisation are given.

2.3.3 Ant colony optimisation

Ants cannot see and yet have the amazing ability to find the shortest path from A to B. Starting from A, ants walk around randomly until they reach their target B. During their journey, the ants lay some pheromone, marking their paths. The pheromone slowly evaporates. On their way back to A the ants follow the pheromone trails and reinforce them with new pheromone. After a while, ants have a higher probability of taking the shorter routes, because the ants on the shorter trails were able to make more journeys in the same amount of time. Therefore the pheromone on this trail is reinforced more and is stronger than the pheromone on the longer trails [27].

This ant system is an example of the swarm theory; many agents follow simple rules, but together they are capable of showing intelligent behaviour [11]. Dorigo et al. [27] have introduced a computational optimisation algorithm based on the optimising behaviour observed by ants. Artificial ants are capable of exploring a multidimensional search space and finding an optimum solution [8]. Ant colony optimisation will be discussed further in Chapter 3.

2.3.4 Evolutionary structural optimisation

By slowly removing inefficient material from a structure, the shape of a structure evolves towards an optimum. This is the simple concept of *evolutionary structural optimisation* (ESO) [19]. Straightforward and effective, ESO is a popular method for structural optimisation. The easiest versions of ESO mesh a volume and reject material based on the stress level due to a load case. The higher the rejection ratio, the more material is removed. In the end only the most effective material remains.

Because of its success and popularity, ESO is well developed and several extensions have been developed. In the concept of *additive evolutionary structural optimisation* (AESO) material is added near highly stressed areas. A combination of both adding and removing material is called *bi-directional evolutionary structural optimisation* (BESO) combines the two and is capable of both [64]. Several more variations to the ESO concept are and are being developed. Instead of taking stress level as a decision criterion, one can also choose to take frequency, buckling or other criteria to base the rejection decision upon [19].

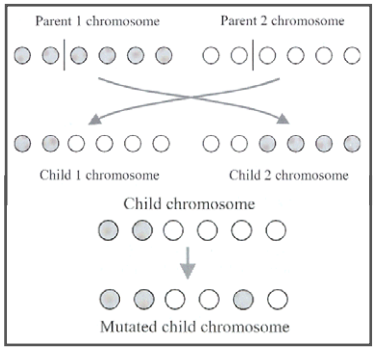


Figure 2.10: Visual representation of GA. Pictures from Bentley [2].

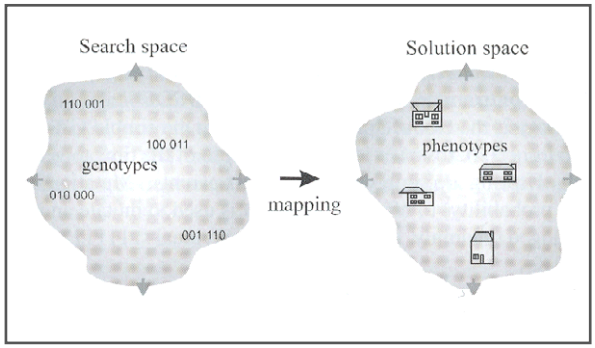


Figure 2.11: Visual representation of separate search and solution spaces, linked by mapping. Picture from Bentley [2].

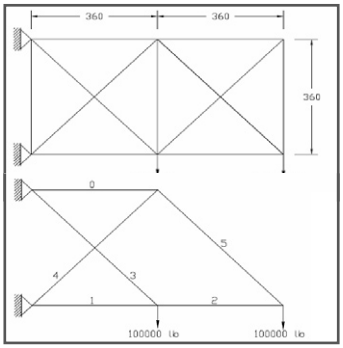


Figure 2.12: GA optimisation. Picture from Deb and Gulati [25].

2.4 Application in structural design

2.4.1 Mapping

The optimisation methods that are discussed in this chapter are not all specifically designed for optimising structures. In order to be able to use either Lagrange, Monte Carlo, differential evolution, simulated annealing, genetic algorithms or ant colony optimisation on a structural design problem, one needs to represent the structure in a clever way, so that it is suitable for optimisation with any of those concepts.

An interesting finding concerning this problem is that in genetic algorithms, this challenge is conquered using two different spaces; a search space and a solution space. Mapping the solution to a code, and the code to a solution, links the two spaces [2]. In order to use any of the concepts mentioned above in structural optimisation, mapping will be necessary.

2.4.2 Local-global problems

In the mathematical theory of optimisation one of the biggest problems is the risk of getting stuck in local optima. The objective function mostly possesses more than one optimum with a derivative of zero. In other words, if one finds the top of a mountain or the bottom of a valley, it isn't necessarily the highest top or the lowest valley [5].

Several optimisation techniques are sensitive to getting stuck in local optima, and avoiding this is one of the major challenges in optimisation. Especially in computational iterative optimisation, the phenomenon of getting stuck in a local optimum is called *stagnation*.

2.4.3 Types

Optimisation in structural design can have different purposes. The major difference lies in the level of detail in the optimisation results. In this thesis the types of optimisation as listed by Coenders will be used [5], and are described here in order of growing detail.

Topology optimisation

The purpose of topology optimisation is to find the optimal layout of a structure within a specified domain. Any solution can arise. The only boundary conditions are the applied forces and the support reactions, and the domain in which structural elements can be placed [5].

Shape optimisation

Shape or geometry optimisation alters a given geometry to an optimum solution. The geometry is modified during the optimisation process but does not change to a different type. Performance is maximised by changing thicknesses in certain places, changing curvatures or adjusting node locations [5].

Size optimisation

With the type and shape of the design fixed, the choice for individual members within the structure can be optimised. Cross sectional properties are used as a design variable. This type of optimisation is referred to as size optimisation [5].

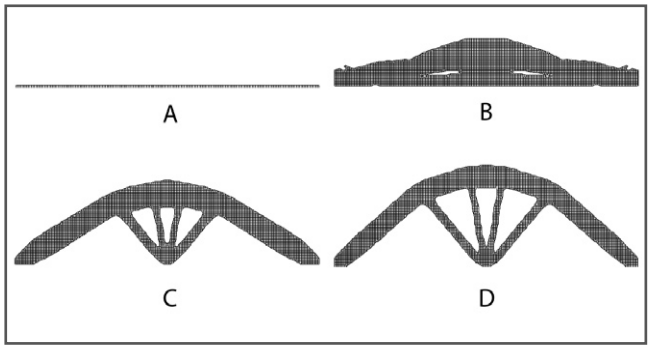


Figure 2.13: BESO performed on a pinned girder. The result resembles half the structure shown in Figure 2.6. From Huang et al. [36].

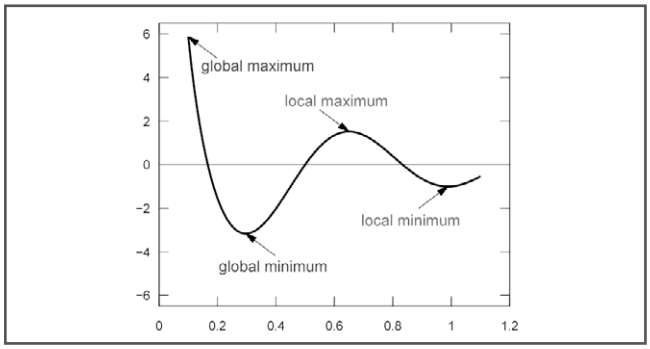


Figure 2.14: Local and global minima and maxima. Picture from Wikipedia [71].

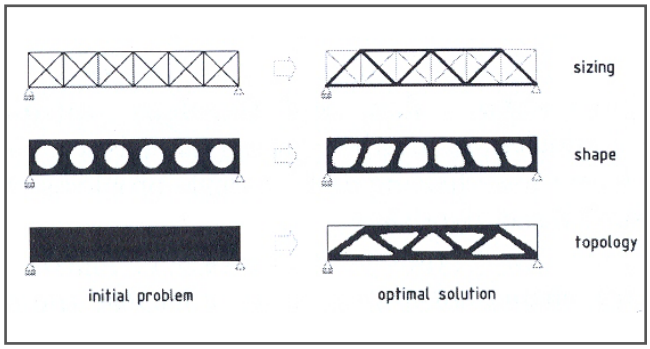


Figure 2.15: Different types of optimisation. Picture from Coenders [5].

CHAPTER 3 ANT SYSTEM AND ITS APPLICATIONS

3.1 Real ant behaviour

A colony of social insects like ants behaves like a superorganism. Ants cannot see and have no overseer or ruler in the colony. Each worker has access to only very limited local information. Yet as a colony they are capable of efficiently searching a space, selecting the richest food source and finding the shortest path to get there [4, 26, 30].

The behaviour of ants has been researched thoroughly. The exploratory behaviour of the Argentine Ant *(Iridomyrmex Humilis)* is described by Goss et al. [30] and Deneubourg et al. [26], while Camazine et al. [4] focus on the Black Garden Ant *(Lasius niger)*. The foraging behaviour of these different types of ants when finding a shortest trail to their food source is similar.

3.1.1 Positive feedback

When looking for food, individual Black Garden Ants move around randomly. When they happen to find a food source, they will mark their trail on the way back to the nest. For this they lay a chemical substance called *pheromone*. The ants control the amount of pheromone they deposit. When the source found is rich, they deposit more than when the source found is poor. The pheromone gets gradually weaker after being deposited and has a lifetime of 30 to 60 minutes [4].

The pheromone trails can be detected by other ants, and the chance they will follow a trail depends on the amount of pheromone deposited on the path. The amount of pheromone determines the probability that ants follow a certain pheromone path. This gives a positive feedback to the trail, because the ants that choose the trail will reinforce the pheromone on it with their own. Now more and more ants will follow the trail. Still, some ants get lost or look for trails outside the explored paths. They might find shorter paths or richer food sources. Whenever they do so, their paths will be intensified quickly, as will be explained later [4].

While most ant species show the same behaviour in trail laying, the Argentine ant has a slightly different approach. This species lays its pheromone more or less continually whilst exploring and not only when a good food source is found [26]. This difference in pheromone laying and thus in optimising behaviour is interesting because a similar difference can be made within the ant system, as will be explained later in this chapter.

3.1.2 Double bridge experiments

The converging behaviour of ants towards the shortest trail will be explained with the findings of the experiments performed by Goss et al. [30]. In the experiments, Goss et al. connect an ants nest and a food source by a double bridge. The ants have to pick one of the branches of the bridge. Initially, their choice is random. The trail they lay on the bridge branch attracts other ants. Now suppose one branch is half

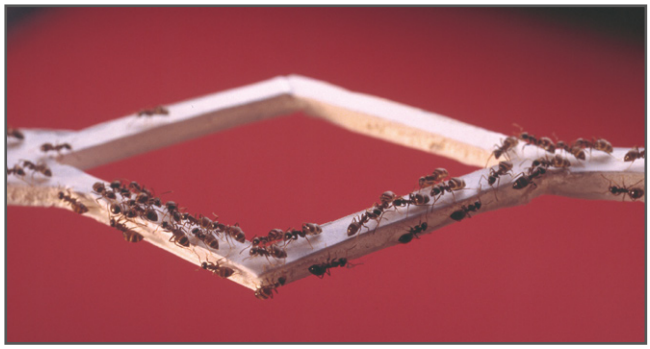


Figure 3.1: Ants converge to one out of two equally good solutions. © CNRS. Reprinted with permission.

Figure 3.2: Ants tend to choose the shortest route over a double bridge. © CNRS. Reprinted with permission.

the length of the other. This means that in a certain amount of time, ants that pick the short branch will make twice as many crossings as the ones picking the long branch. Hence, they will be able to lay twice as much pheromone. After a while, therefore, the pheromone on the short branch will be stronger than the pheromone on the long branch and ants will have a higher probability of choosing this path. This higher probability will further reinforce the pheromone on the path and the ants will converge to the shortest path [30].

When the experiment starts all over again on a bridge with two branches of equal length, the ants select with the same probability any of the branches. Yet, because of random fluctuations, a few more ants will select one branch over the other. Because ants deposit pheromone while walking, a larger number of ants on a branch results in a larger amount of pheromone on that branch. This larger amount of pheromone in turn stimulates more ants to choose that branch again, until finally the ants converge to one single path [30]. It is important to remember that this process of positive feedback is the basis of the optimisation technique ants use when selecting a solution [8].

An interesting experiment is conducted by Goss et al. [30] is to see what happens when – after converging – the ant colony is offered a new and shorter connection between the nest and the food. It was found that the ants stick to the long branch and the shorter branch is only used occasionally. This can be explained by the high concentration of pheromone on the long branch. The great majority of ants choose the long branch because of the high concentration of pheromone. The positive feedback behaviour continues to reinforce the long trail, so that the ants are stuck in a non-optimal solution [30]. Quicker evaporation of the pheromone could favour the exploration of new paths for it would allow the ant colony to *forget* the suboptimal path to which they have converged, but is too slow in these real ant experiments [8]. This is an important observation to keep in mind when developing an optimisation routine based on the ants' foraging behaviour.

3.2 Ant system and its extensions

The double bridge experiments show that ant colonies have a built-in optimisation capability. They use local information for probabilistic rules that help them finding the shortest trail between two points. In more abstract terms: ants can search a space and find a global optimum with limited information. The double bridge experiments also show under some circumstances ants will get stuck in a non-optimal solution.

This optimising behaviour has been the inspiration for a family of computational algorithms. This family started with the *ant system* (AS) [27]. Several extensions have been proposed which led to the definition of a framework that applies to all the different ant system based algorithms. This framework is called *ant colony optimisation* (ACO) [8].

3.2.1 Ant system

The first internationally published proposal for a computational search algorithm based upon the optimising behaviour of a colony of ants is the ant system [27]. This algorithm is primarily applied to the *travelling salesman problem* (TSP).

Travelling salesman problem

An easy-to-understand benchmark for the ant system is the travelling salesman problem. In this problem a salesman, starting from his hometown, wants to find the shortest tour that takes him through a given set of customer cities and then back home, visiting each customer city exactly once.

Simple in concept, the number of possible solutions for this problem is $\frac{1}{2}*(n-1)!$ for $n>2$, where n is the number of cities. TSP is classified as a NP-hard problem, which means that the amount of time needed for exact algorithms to find the optimum is in the worst case exponentially dependent on the size of the problem [8]. In order to solve travelling salesman problems where many cities need to be visited, algorithms that give an approximate solution are often used. Such algorithms are also called *heuristics*.

Artificial ants

In order to solve the TSP, artificial ants are introduced in the ant system. Dorigo et al. [27] explain that each ant is a simple agent with the following characteristics.

- It chooses the town to go to with a probability that is a function of the town distance and of the amount of pheromone present on the connecting path.
- To force the ant to make legal tours, transitions to already visited towns are disallowed until a tour is completed, using a taboo list.
- When it completes a tour, it lays a substance called pheromone on each path visited, which is a function of the tour length.
- Pheromone evaporates over time.

Two major differences with real ant behaviour can be observed in the way the ant system solves the travelling salesman problem. First of al, the problem of the travelling salesman is made discrete. Every move from one city to another is done in one iterative step. This means that finding a shorter route does not give the ant the possibility to go back earlier, reinforce his own pheromone trail and attract more others to his trail. Instead, distance is measured and saved, and the pheromone is distributed according to the performance of the trail, which depends on its length.

Secondly, the artificial ants used in AS are not completely blind. They take the distance to the next city into account when deciding where to go. This makes them greedy; rather than moving around randomly, they tend to achieve short-term targets.

Control parameters

Controlling the pheromone on the paths is essential for an efficient ant system.

- Pheromone evaporation makes it possible to forget suboptimal paths.
- Assigning more pheromone to paths with better performance encourages quick convergence and brings along the risk of premature stagnation.

These control parameters play a role in the behaviour of real ants, as well. In the double bridge experiment where a shortcut is offered after convergence to the long branch, the

amount of existing pheromone leads to ants sticking to the suboptimal solution. Quicker evaporation of the pheromone would avoid this. When one considers the different kinds of ants, it is observed that the Argentine Ant constantly lays pheromone, while the Black Garden Ant only lays pheromone on the trail that has proven to be good. This is a similarity to the assignment of more pheromone to better solutions, as is done in the ant system algorithm.

The results found for the ant system as presented in the paper by Dorigo et al. [27] for the TSP were slightly better but comparable to other algorithms based on natural phenomena like simulated annealing and genetic algorithms [27]. The ant system is versatile, robust and population based, allowing the exploitation of positive feedback as a search mechanism. All these properties are considered desirable for a heuristic, and encouraged the extension of the ant system in other algorithms.

3.2.2 Extensions of the ant system

Several ant system based algorithms have been developed after its original introduction. The most important difference between the different ant system based algorithms is the way pheromone is assigned to the paths. As stated earlier, the way the pheromone is laid and evaporates is very important in the trade-off between the exploratory behaviour of the algorithm and the speed of convergence toward an optimum.

As an example, the *MIN-MAX* Ant System (*MM*AS) is discussed, introduced by Stützle and Hoos [55]. *MM*AS copes with pheromone in a very different way than the AS. There are three major differences between the two.

- Only the best performing ant adds pheromone to the paths.
- The minimum and maximum amounts of pheromone are explicitly limited.
- In the first iteration, every path contains the maximum amount of pheromone.

Because of these adjustments, the search space will be searched more thoroughly and local optima can be avoided [55].

Many other ant system based algorithms, like the *elitist ant system*, the *rank-based ant system* and the *ant colony system* all have the same property of assigning more pheromone to the best trail, or the best set of trails, than the original ant system does.

It is noted that in real ant behaviour, more pheromone is assigned to shorter trails because the ants travelling on them go back and forth more often than the ants on the longer paths do. Computational ant systems try to obtain the optimising behaviour in a different way, making the reinforcement of the pheromone a rather mathematic reflection of the performance.

3.2.3 The ACO metaheuristic framework

For all the different ant system based algorithms, a framework called *ant colony optimisation* (ACO) has been proposed [8]. Algorithms that fit into this framework are called ACO algorithms. The general properties of ACO are discussed in this paragraph.

Metaheuristics

Heuristic algorithms are algorithms that give an approximate solution to a problem. They seek to obtain good, that is, near-optimal solutions at relatively low computational cost. They are not guaranteed to find the one optimal solution. Especially when a problem has many possible solutions it is not efficient to check every single solution. Heuristics trade optimality for efficiency [8].

A *metaheuristic* is a general algorithmic framework which can be applied to different optimisation problems. If the metaheuristic is taken as a starting point only a few modifications are necessary to apply an algorithm to a specific problem [8].

Problem representation

Artificial ants incrementally build solutions by adding solution components that are chosen based on a distribution of probabilities. Given this behaviour, there are some boundary conditions to the way a combinatorial optimisation problem is represented in order for the ants to be able to solve it. The mapping of the considered problem to such a representation is the real issue in ant colony optimisation.

In the minimisation problem (S, f, Ω), where S is the set of candidate solutions, f is the objective function which assigns a value $f(s,t)$ to each candidate solution $s \in S$, and $\Omega(t)$ is a set of constraints. The parameter t indicates that the objective function and the constraints can be time-dependent. The goal is to find a global optimum s^*, that gives a minimum cost.

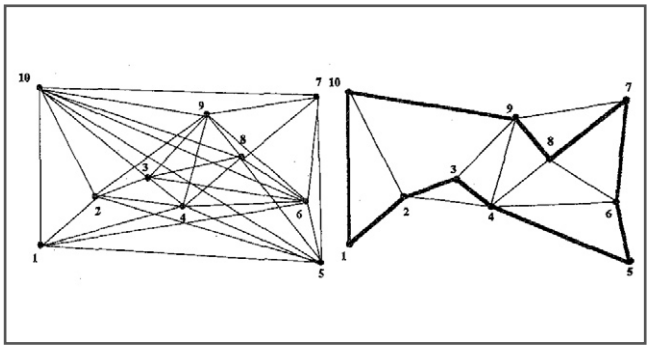


Figure 3.3: A 10-point TSP problem. Pheromone distribution at the beginning and after 100 cycles. Picture from Dorigo et al. [27].

```
procedure ACOMetaheuristic
    ScheduleActivities
        ConstructAntsSolutions
        UpdatePheromones
        DaemonActions                % optional
    end-ScheduleActivities
end-procedure
```

Figure 3.4: The ACO metaheuristic in pseudo-code. From Dorigo and Stützle [8].

The combinatorial optimisation *(S, f, Ω)* problem can be mapped to a problem having the following characteristics [8]:

- A finite set $C = \{c_1, c_2, \ldots, c_{N_C}\}$ of components is given, where N_C is the number of components.
- The states of the problem are defined in terms of sequences $x = \langle c_i, c_j, \ldots, c_h, \ldots \rangle$ of finite length over the elements of *C*. The set of all possible states is denoted by *X*.
- The set of solutions *S* is a subset of *X* ($S \subseteq X$).
- A problem-dependent test verifies whether boundary conditions are met and defines a set of feasible states $\tilde{X}$ with $\tilde{X} \subseteq X$.
- A non-empty set *S** of optimal solutions, with $S^* \subseteq \tilde{X}$ and $S^* \subseteq S$.
- A cost is calculated for each candidate solution $s \in S$, mostly by *f(s,t)*.
- In some cases a cost *J(x,t)* can be associated with states that are no candidate solutions.

Ants' behaviour

Given the formulation, artificial ants build solutions by performing randomised walks on the completely connected graph G_C=*(C,L)*. The nodes are the components *C* and the set *L* fully connects the components. The ants as a colony have a memory which is formed by the pheromone trail. Each ant selects a move by applying a probabilistic decision rule. The probabilistic decision rule is a function of the locally available pheromone trails and heuristic values, the ant's private memory storing its current state and the problem constraints. The ant can update the pheromone associated with the connection it chooses while walking, or it can retrace his path and update the pheromone once it has built a complete solution [8].

The ACO metaheuristic

An algorithm can be represented in pseudo-code, the most condensed code of an algorithm. The pseudo-code of ACO metaheuristic, as introduced by Dorigo and Stützle [8], is given in Figure 3.4. ACO algorithms consist of three procedures.

The ConstructAntsSolutions procedure contains the stochastic local decision policy and lets ants incrementally build solutions to the problem. The UpdatePheromones procedure evaluates the solutions and distributes pheromone on the used path accordingly. DaemonActions is an optional procedure that implements centralised actions which cannot be performed by single ants. For example, the daemon can allow the best few ants to lay additional pheromone on their trail.

3.3 Applications in structural design

ACO algorithms have been developed for many problems. The vast majority of them has a direct link to minimising path lengths. Routing, task assignment, scheduling, subset, machine learning and network routing problems are examples of areas for which ACO algorithms are developed [8].

For application in structural design, it is important to look at the problem in a more abstract way. One should not focus on minimising trail lengths, but rather on combining properties in different parts of the structure and evaluating the result. A few examples of application of ACO in structural design are done by Camp et al. [23, 24], Samdani et al. [51] and Serra and Venini [52]. All of the problems considered are rather simple discrete shape optimisation problems of steel truss design. The position and the connection of the elements are given, but the cross section of the member can be changed in the optimisation process.

The interesting thing about these algorithms for truss optimisation is the way the problem is represented. The problem is represented as a minimum path problem by modelling the steel volume as a path length. Each different cross section that can be chosen for a member is represented by a path; more material means a longer path. The artificial ants walk on each line that represents a member, choosing one of the sizes. They do not necessarily make a physically possible route. Instead, the members are mapped to a circle which the ants walk. All this is explained in Figure 3.5. By mapping the structural optimisation problem this way, a search space is connected to a solution space.

Camp et al. find solutions that are comparable to those found with genetic algorithms, but he needs more iterations to come to these solutions. He notes that ACO through the use of pheromone trails uses information from all previous iterations and that ACO is less sensitive to poor initial solutions. He concludes that ACO has a great potential for solving constrained discrete problems [23].

Recently, some attempts have been made to apply ACO to topology optimisation in structural design. Luh et al. [42] propose topology optimisation based on ACO. In their approach, an artificial agent is placed on a mesh that

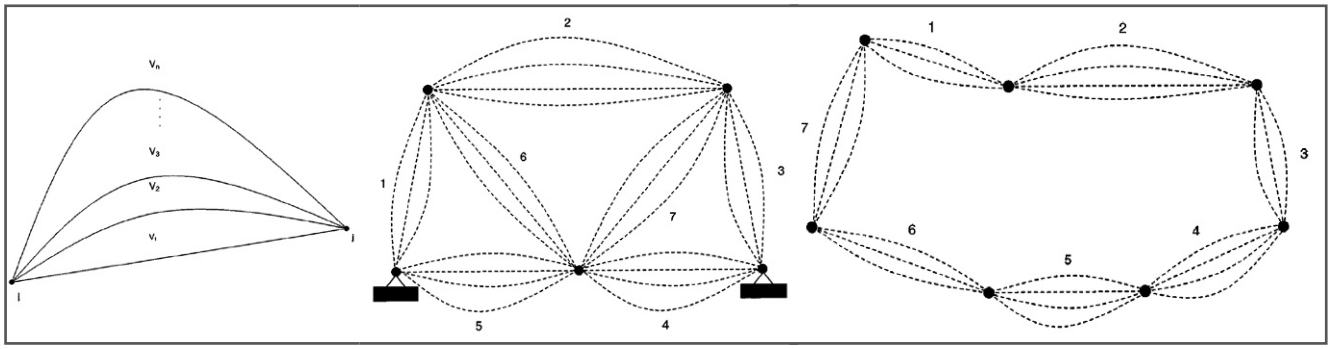


Figure 3.5: Representation of the truss member shape optimization problem. Pictures from Bornemann and Faber [23].

represents the space available for structural elements. The agent is allowed to move to adjacent elements only, and can only make a limited amount of moves during an iteration. This way a solution is construced by several ants together. The concept of mapping is used in this approach as well as an elitist system. Luh et al. derive a family of near-optimal structural topologies from the pheromone distribution that the ants have generated. The designer is left with the task to choose one solution out of the pool. Luh et al. explicitly state that the resulting shapes are fairly poor and can only be used as an inspiration and are not an optimum themselves [42].

A different approach is shown by Mitsui [45]. He starts with a feasible but non-optimal solution and uses only a local rule to adjust this initial solution and converge to an optimum [45]. Though his results are interesting, the algorithm is merely inspired by ants' behaviour and does not fit in the ant colony optimisation metaheuristic framework as defined by Dorigo and Stützle [8].

3.4 Conclusions

Ant colony optimisation algorithms, or ant system based algorithms, are able to solve combinatorial optimisation problems. Several variants to the original ant system are all referred to with the term *ant colony optimisation*. They differ in the way they cope with the distribution of pheromone. However, mapping a problem to the specific representation needed for ACO is the major challenge.

Some applications of ACO in structural design are known. Most apply to choosing sections in a truss construction; a very discrete problem. Given the nature of the algorithm, ACO can be applied to topology optimisation if a right formulation and representation for the problem is found. Two research projects on this topic were found. They have a different approach. Both limit the exploratory behaviour of the ants by strict boundary conditions, because of which only feasible structures are generated.

CHAPTER 4 ULTRA-HIGH PERFORMANCE CONCRETE

4.1 Overview of innovations in concrete

Starting with mud, straw and wood, building materials have evolved over the years with the knowledge of mankind. After stone blocks had gained common use, the Minoan around 2000 BC figured out a way to glue blocks together using a lime mortar. The Greek copied this technology. However, lime mortar dissolves in water and therefore is not very durable. It were the Romans who managed to invent a more durable material that we nowadays know as concrete [58].

4.1.1 Vitruvius' invention

It was around the Vesuvius volcano, near the Italian town of Puzzoloni, where Vitruvius found the volcanic ash that he used to produce the first ever concrete. As he said it in his world-famous book *Architectura* [16]:

Hence, when the three substances […] are mixed together, the water suddenly taken in makes them cohere, and the moisture quickly hardens them so that they set into a mass which neither the waves nor the force of the water can dissolve.

Materials that react with water and lime to a hard substance are called pozzolans, named after the town where Vitruvius discovered the volcanic ash. Pozzolans are cementitious materials. Vitruvius mixed his pozzolanic volcanic ash with lime and gravel, which is pretty similar to the non-reinforced concrete that is still used today.

The new building material that was invented led to new structural shapes. The Romans invented the arch, the vault and then the dome. They built aqueducts for transportation of water and showed their knowledge in the structure of the Pantheon. Spanning 43.3 metres in circumference, this structure remained the biggest dome in the world until 1910 [58].

4.1.2 Reinforced concrete

Concrete technology did not really develop after the Roman times. Englishman Joseph Aspdin introduced the Portland cement in 1824 giving a new boost to concrete development. However, is was the Frenchman Joseph Monier in 1850 who discovered that he could construct very strong flowerpots using an iron mesh and mortar. This was the introduction of reinforced concrete, a new material that could not only

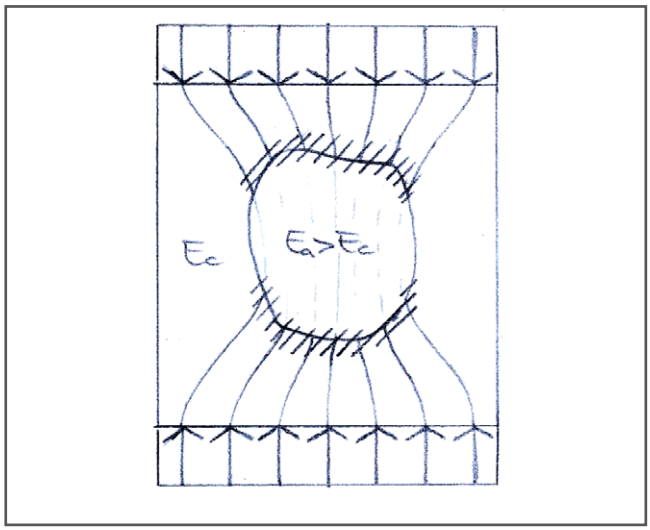


Figure 4.1: Internal peak stresses due to coarse aggregate.

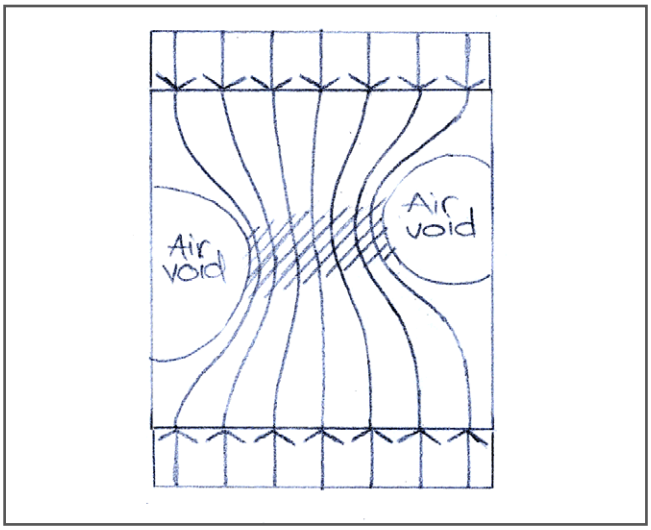


Figure 4.2: Internal peak stresses due to air voids.

Peak stress in the cement paste

cope with compressive, but also with small tensile stresses and thus with bending. Monier's fellow Frenchman François Hennebique is responsible for the first success of reinforced concrete in the building industry. He developed a floor system for buildings. As in Roman times, the new material led to new structural concepts [14].

At the turn of twentieth century concrete characteristic compressive strength was around 15 MPa. This developed to 25-45 MPa in the sixties [13]. Even today, characteristic strengths of around 35 MPa are very common for concrete. Pre-fabricated elements can be stronger (characteristic strength about 65 MPa) and can harden more quickly because the environment in which they are cast and cured is better controlled.

Pre-stressed concrete

Adding reinforcement to the concrete solves the problem of the brittleness and improves the concrete's performance in tension. However, its capacity in tension is several times lower than its capacity in compression. Pre-stressing the concrete solves this problem. By applying a compressive force to the concrete member, tensile forces or bending forces only lead to tension in the member when the stress gets larger than the compressive pre-stressing.

Fibre reinforced concrete

Rather than distributing the reinforcement of the concrete discretely, as is done when conventional reinforcement bars are used, fibre reinforcement is cast with the concrete and therefore distributed randomly. The small fibres come into action when microcracks occur and prevent macrocracks from developing. Their size, especially when used for concrete with superlative strengths, is mostly ranging from 13 to 40 mm in length and 0.1 to 0.3 mm in diameter [20, 31, 33, 56]. For cases like industrial floors, larger fibres can be used.

4.1.3 Superlative strengths

The introduction of high-range water reducers in the early nineties made it possible to use a lower *water/cement ratio* without the loss of workability. This caused a revolutionary increase in concrete strength. A new concrete was developed that had a characteristic strength of 200 MPa, which was a huge improvement compared to the 65 MPa that was a practical upper limit at the time. This improvement was possible because of the use of very small aggregates like silica fume, the use of fibre reinforcement and very low water content in combination with the use of high-range water reducers [10, 48, 60].

Nowadays this kind of concrete is known as *ultra-high performance concrete*. This is a very subjective term, but as in most of the literature, in this thesis this term is used for concrete with a characteristic compressive strength of at least 150 MPa, a ductile behaviour due to the systematic use of fibres, a high binder content and a special selection of aggregates resulting in self-compacting behaviour [1, 43, 44, 48, 60]. As with the earlier development of concrete and reinforced concrete, the development of this new material gives the chance to again come up with efficient and new structural concepts.

4.2 Obtaining UHPC

A concrete with a compressive strength of 200 MPa was first applied in the case of the Sherbrooke footbridge. This was a huge step forward from the concrete used so far, having an absolute maximum strength of 100 MPa [60]. Optimising the concrete mixture for strength only, however, neglects the need for workability. Grünewald [62] shows that there is a huge possibility of making very workable mixtures in the range of 100 to 200 MPa. In other words, the trade-off between strength and workability still exists. Now, when strengths as high as 800 MPa are observed in laboratories, it is important to realise that if one wants a constructible building in UHPC, that has its consequences on the strength. Further to strength and workability, one can also optimise with respect to sustainability, durability, affordability et cetera.

4.2.1 Concept

When designing a UHPC mixture, two things are of major importance; reducing internal peak stresses and avoiding brittle behaviour of the hardened concrete.

The reduction of internal peak stresses is obtained in three ways.

- The used aggregate is small and strong. Coarse aggregate would cause internal differences in elasticity and thus peak stresses in their contact areas and tensile stresses in the cement paste to keep the aggregate together.
- The spaces in between the cement particles are filled with small reactive powders like silica fume. This increases the

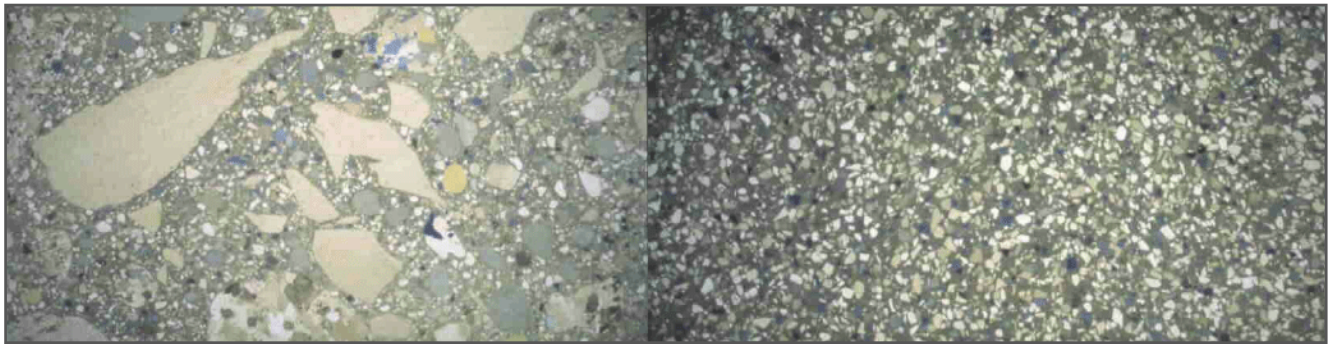


Figure 4.3: Comparison of ordinary concrete and UHPC on the same scale. Picture from Resplendino [49].

contact area in between the particles and hence reduces the stresses.
- A very low water/cement (w/c) ratio is used to avoid air voids and the peak stresses around them.

Brittle behaviour of the hardened concrete is avoided by adding fibre reinforcement [10].

4.2.2 Mixture properties

Water and cement

The chemical reaction that occurs when water and portland cement are mixed is the basic principle behind all concrete. The reaction creates *calcium silicate hydrate*, also referred to with the terms *CSH gel* or *cement paste*. Portland cement is made of lime, silica and alumina. After the ingredients under heating and mixing have formed so-called *clinkers*, the cement is ground to small particles with an average size of 10 µm [10, 13, 29, 43].

Water is a crucial ingredient of the concrete mixture. It has four effects. In the first place, water is needed to react with the cement. Secondly, water is needed to lubricate the mixture for a better flow of the mixture. This means an improvement of workability. Easy flow of the mixture reduces the entrapment of air and voids resulting from this entrapment. Thirdly, evaporation of the excess water that does not take part in the chemical hydration process causes voids. Additionally, contaminated water can contain harmful ingredients and can be destructive to the strength of the cement paste [13].

Peak stresses occur around voids. In order to reduce the peak stresses a very low water/cement ratio is used in UHPC. It is better to have some remaining cement in the hardened concrete than excess water, for it can function as a fine filler [60]. In ultra-high performance concrete the w/c ratio can be as low as 0.17 [50]. Hendriks [10] claims that in order for cement to fully react it needs an amount of water of around 25% of its own weight. In ultra-high performance concretes, reactive powders other than cement are used as well. It is now arguably more interesting to measure the water to binder ratio, w/b, which indicates the weight of the water in the mixture compared to the cumulative weight of the binders. Such binders can be cement, silica fume and sometimes fly ash.

Silica fume

The extremely fine and reactive silica fume, sizing from 0.1 to 1 µm [29, 43], fills the spaces between the cement and fly ash particles. The best packing is obtained if the ratio of silica fume weight over cement weight in the mixture is about 0.25 [43, 50]. Its fineness and therefore its high surface area lead to an increase of the water demand up to 40% [13]. The water to binder ratio has an optimum in between 0.13 and 0.15 [50].

Because of the considerable increase in water demand and therefore the reduction in workability techniques are being developed to attain the same quality of concrete using less silica fume [40].

High-range water reducers

The use of *high-range water reducers* (HRWRs) or *superplasticizers* is essential in UHPC. They distribute the cement and silica fume particles evenly keeping them from making conglomerating balls. This improves workability. Now similar workability can be obtained with a lower w/c or w/b ratio. In UHPC mixes the amount of HRWR is about 10 to 20% of the weight of the water [13, 32, 43].

When a mixture flows easily the entrapment of air is diminished, which otherwise could cause major air voids [13].

Fly ash and ground granulated blast furnace slag

Fly ash and ground granulated blast furnace slag (GGBFS) are small pozzalanic particles that improve the density of the concrete. Just like silica fume, the particles fill up the spaces in between the cement particles. Because of the pozzolanic property of fly ash and GGBFS, they react to a CSH gel. Because the size of the particles is not quite as small as silica fume particles, their effect on the density of the concrete is not as dramatic as the effect observed when silica fume is used [13].

Quartz

Finely ground quartz with a diameter of 1 to 10 µm is frequently used for optimisation of the packing density as an addition between the grain size distributions of micro silica and cement [43]. By grinding quartz to a different diameter, it is a useful additive for the grain size between cement and sand [28].

Aggregates

To reduce peak stresses, small aggregate should be used in concretes that are designed for high strengths. In *reactive powder concrete* (RPC), introduced in the nineties by Richard and Cheyrezy [50] the biggest aggregate is quartz sand up to 0.8 mm. In UHPC, the biggest aggregate is around 5 mm [43,

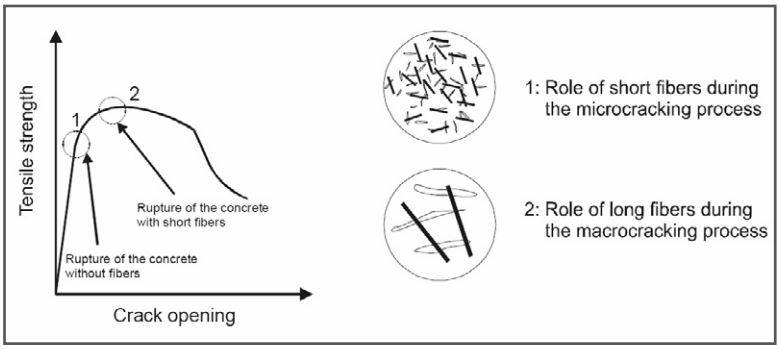


Figure 4.4: Role of short and long fibres during the cracking process. Picture from Orgass and Klug [46].

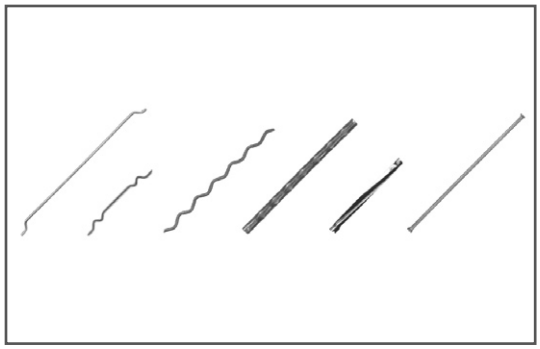


Figure 4.5: Different kinds of fibres. Picture from Weiler and Grosse [61].

46]. The aggregate should be of at least the same strength as the cement paste.

Due to the high quality demands, the aggregate is more expensive than cement. UHPC therefore typically has a low aggregate content.

Fibre reinforcement

In general, fibre reinforcement is used in large amounts in UHPC [48, 60]. Fibre reinforcement can cause hardening of the concrete observed in the stress-strain relation; after microcracks have developed, the tensile stress can increase further [13, 60].

Several kinds of fibres can be used. Orgass and Klug [46] give the following overview of types of fibres and their effect on the properties of the concrete.

Steel Fibres
- Increase of fracture energy, subsequent improvement of ductility.
- Increase of strength (compressive strength, tensile strength).
- Reduction of tendency for cracking.

Polypropylene Fibres (PP fibres)
- Decrease of microscopic crack growth with high loading.
- Gain in fire resistance.
- Decrease of early shrinkage.

Glass Fibres
- Reduction of internal stresses within young concrete.

To increase the ductility and to improve the fracture toughness of the concrete a cocktail of different lengths of steel fibres can best be introduced [46, 60]. Short steel fibres help to increase the mechanical properties, because they extend the elastic range and fit best in the fine matrix in the concrete [46]. Additional polypropylene fibres reduce the evaporation of water in young concrete and consequently reduce shrinkage and crack width improving durability. These fibres also melt quickly in case of fire, producing a channel system in which liquid and steam can expand. This improves fire resistance [46].

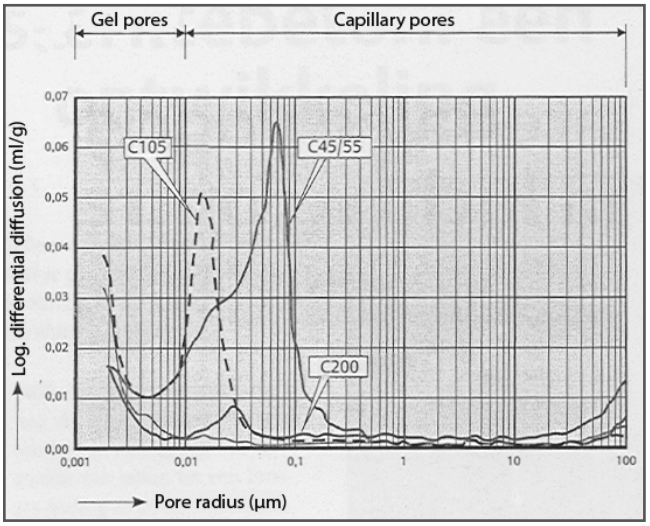


Figure 4.6: Pore radii in different kinds of concrete. Picture based on Walraven [60].

An adequate mix of fibres is essential in UHPC, normally comprising around 2% of the concrete volume and sometimes more. It is important to keep in mind, though, that fibre content is the most important parameter for the cost of the concrete, and that fibres reduce workability [13, 46, 60].

A homogenous distribution and orientation of fibres is often assumed but rarely obtained. Inhomogeneity is caused by the direction of flow during concrete placing, natural alignment parallel to the formwork and gravitational preference of the fibres [48, 62].

4.2.3 Curing

In situ and pre-cast concrete

Further to the strength/workability trade-off, it is possible to pre-fabricate concrete elements in a factory. In a controlled environment strength development can be controlled better and workability issues can be dealt with more easily. It is therefore possible to obtain a better concrete quality when elements are pre-fabricated in a factory.

Heat treatment

The microstructure of concrete can be improved through hot curing. The most common method is curing the concrete at 90˚C for two days [13, 35]. This speeds the activation of the pozzolanic reaction of the silica fume. This results in the earlier obtaining of the final strength, decrease in shrinkage after the heat treatment to zero, reduction of creep and a gain in compressive strength of 10 to 50% [1, 13, 56].

Telebinejad et al. [56] experimented with more extreme heat treatment methods, obtaining a huge 114% gain in compressive strength by curing the concrete at 180˚C. The gain can be explained by the formation of *xonolite*, a special type of CSH gel, in such high temperatures. They also prove a more rapid strength development due to the heat treatment.

Heat treatment is an expensive process and means that the mould cannot be re-used until heat treatment is completed, slowing down production.

Compression curing

The density of concrete can be increased by maintaining the fresh concrete under pressure at the placement stage and during setting. Air bubble reduction, expulsion of excess water and partial reduction of the plastic shrinkage during the final set are some of the results. Compression curing asks for a perforated mould and is very difficult for large elements [50].

	ordinary concrete	UHPC
Water porosity (%)	14 - 20	1.5 – 5
Oxygen permeability (m^2)	10^{-16}	$<10^{-19}$
Chloride-ion diffusion (m^2/s)	$2*10^{-11}$	$2*10^{-14}$
Portlandite content (kg/m^3)	76	0

Table 4.1: Durability indicators. From Resplendino and Petitjean [48].

4.2.4 Performance characteristics

Early age properties

Due to the high strength of UHPC, the early age strength is much higher than ordinary concrete, as well. Strength development can be accelerated using heat treatment [13, 56].

Workability

UHPC is often self-compacting, in which case it does not need to be vibrated after casting. The small-sized ingredients form an easily flowing mixture. HRWRs provide the opportunity to combine a high cement and silica fume content with a low w/b ratio, without affecting the workability. The major parameter affecting the workability negatively is the fibre content, because the fibres diminish the ability of the mixture to flow [32, 46, 62].

Traditional durability issues

The basic principle behind UHPC is the very dense packing. This leads to low permeability. In Figure 4.6 one can observe that UHPC contains less pores and that the pores are smaller than in conventional concrete. Traditional durability issues like sulphate penetration, alkali silica reaction, freeze/thaw, carbonation and corrosion of reinforcement are related to permeability and therefore not a problem in UHPC [48].

The values Table 4.1 are obtained from the preliminary French recommendations on UHPC [1] and are indicators for these traditional issues.

Durability issues upon UHPC

The French organisation AFGC-SETRA did extensive research upon four possible issues specific for UHPC [1].

The first issue is the stability of chemical admixtures that are used in big proportions in UHPC, but only slight surface dissolution has been observed [48].

Possible long-term internal swelling of concrete with residual cementitious particles is not a problem, because water cannot penetrate the concrete. Advantageously, capillary condensation and formation of hydrates causes microcracks to close.

Thirdly, UHPC is particularly effective at maintaining the pH level necessary for passivation of the steel reinforcement and resist to chemical conditions that normally would cause problems. Steel fibres will only corrode at the very surface and not cause any problems except from very light surface staining with only aesthetic consequences, that can be solved by applying a thin coating.

Finally, UHPC provides a good degree of protection against detoriation of polymer fibres due to oxidation or ultraviolet light [1, 22, 48].

Fire

Spalling is the biggest problem when concrete is exposed to a fire. Excess water inside the concrete expands and pushes off pieces of concrete [10]. In UHPC the water content is minimal. The water that is available, however, cannot expand freely because of the dense microstructure and will develop high stresses more quickly. This twofold consequence makes the fire resistance of UHPC a complex problem [34].

The use of polypropylene fibres in UHPC has a positive effect on fire resistance. Above 150˚C they melt and above 200˚C they evaporate, providing additional capillary pores and a channel system as an escape route for the trapped steam [34].

4.3 Types of UHPC

4.3.1 Commercially available pre-mixtures

Several ultra-high performance concretes are commercially available. The mechanical properties of a selection is presented in Table 4.2. As a reference, the original reactive powder concrete as developed by Richard and Cheyrezy [50] is added in the table.

The properties used in the table are all obtained for the same curing conditions without thermal treatment or compression curing. Each manufacturer claims his concrete can obtain higher strengths when such treatment is done. However, in this thesis the concrete will be used on a large scale, which means it is not certain that thermal treatment will be feasible, and that it is likely that compression curing will not be possible.

All mixtures contain steel fibre reinforcement. All mixtures also contain, or can contain with identical mechanical properties, polypropylene fibre reinforcement. They have similar fire behaviour. Non-similar amounts of reinforcement are used in the mixtures. The amount of fibres can be adjusted by the client [32].

			RPC200	BCV	CRC	Ductal® FR	Ductal® US	BSI/Ceracem
Compressive strength	MPa	f_{cj}	170	130	135	150	160	180
Flexural strength	MPa		30	30	20	15	30	30
Tensile strength	MPa	f_{tj}	-	10	10	9	-	8
Tensile post-cracking	MPa	σ_{bt}	0	0	reinf.dep.	-	-	9,1
Young's modulus	GPa	E_{ij}	50	44	44	50	50	65
Density	kg/m^3	ρ	-	2480	3000	2400-2500	2450-2550	2800
Creep coefficient	-	Φ	-	-	-	<0,8	<0,8	0,8
Total shrinkage	-	ε_t	-	<600E-6	-	500E-6	500E-6	700E-6
Thermal expansion	/deg	α_T	-	-	-	11,8E-6	-	10,4E-6
Fibre content	%vol		5-6	2	4	2,15	-	3

Table 4.2: Mechanical properties of UHPCs. Based on [1, 32, 37, 50, 53, 63, 67, 68, 69, 70].

4.3.2 Price

The price of UHPC depends on many factors:

- Ingredients
- Mould surface
- Pre-stress
- Connections
- Shape of the mould
- Building process
- Etcetera

It is difficult to estimate the cost of a concrete structure when the design is still uncertain. Information from experts is used to estimate the price concrete ingredients to be € 2000 per m^3. A rule of thumb for the in-place price of conventional concrete is € 350 per m^3, including material, mould and casting. For UHPC, moulds are probably more complicated because more difficult shapes can be feasible due to the high material costs. As a conservative assumption, a mould price of € 300 per m^2 will be assumed. Additionally, it is assumed for the time being that pre-stressing a cubic metre of concrete to its limit costs € 500. All these values will be reconsidered in Paragraph 6.3.

4.3.3 Choice

In this thesis, BSI/Ceracem will be used. With identical curing conditions and a non-extreme amount of reinforcement BSI/Ceracem gives the best mechanical properties in terms of strength and stiffness. This thesis focuses on future possibilities of applications of this material. Further development in strength and availability of the concrete can be expected and therefore it is wise to choose the strongest concrete available.

The ingredients of one cubic metre of this mixture are as listed in Table 4.3.

Cement	1114	kg		
Silica fume	169	kg		
0-6 mm aggregate	1072	kg		
Fibres	234	kg	3	%vol
Superplasticizer	40	kg		
Water	209	kg	0.19	w/c
Density (sum)	2798	kg		

Table 4.3: Ingredients of 1 m^3 of BSI. From Hajar et al. [32].

4.4 UHPC design properties

4.4.1 French recommendations

The interim recommendations for ultra high performance fibre-reinforced concretes, published in 2002 by the French organisation AFGC/SETRA [1] is the first code-like document on this new material. It is intended to constitute a reference document serving as a basis for use of UHPC in structural applications. The recommendations deal with UHPC characterisation and material properties, design and calculation methods for UHPC and durability aspects [1, 48]. The stress-strain diagrams introduced in these recommendations are particularly interesting [60].

These recommendations are the closest document to a preliminary Euro Code available to date. The structural design methods introduced in these recommendations are used for this thesis.

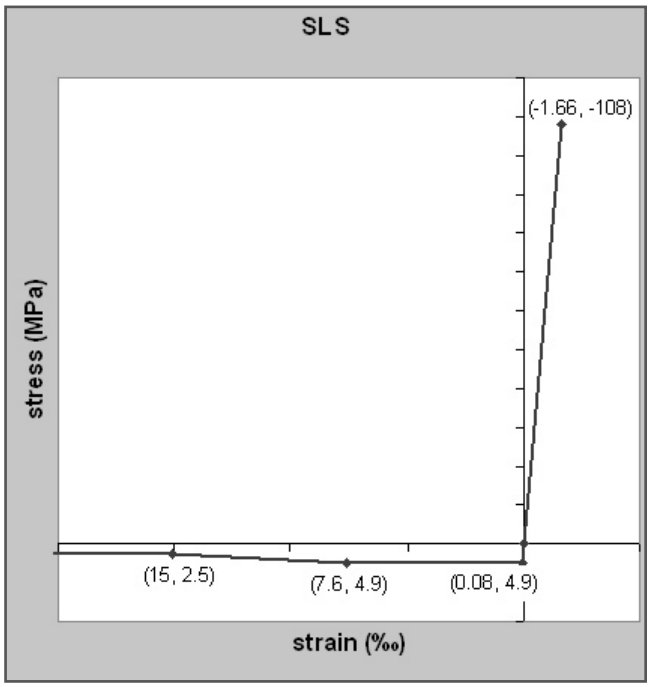


Figure 4.7: Stress-strain diagram for the chosen BSI mixture in SLS.

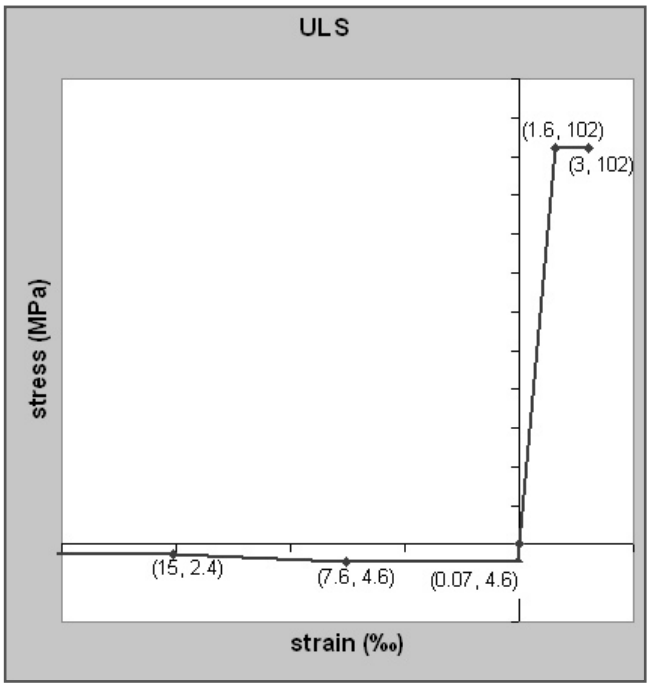


Figure 4.8: Stress-strain diagram for the chosen BSI mixture in ULS.

4.4.2 Design properties for the chosen concrete

Axial stress

For the BSI concrete that is selected in Paragraph 4.3.3, the stress-strain diagrams for both the serviceability limit state (SLS) and the ultimate limit state (ULS) are determined according to the French recommendations. Calculations can be found in Appendix B.1. The diagrams are applicable to fundamental load combinations, global effects and loads that have a duration of 24 hours or more. Though useful for the initial design, the diagrams should be checked once the structural design is completed in order to verify the assumptions.

Shear stress

In the same French recommendations, shear stress verifications for a beam section are given for the serviceability limit state [1].

$$\tau^2 - \sigma_x \sigma_t \leq 0.25 f_{tj}\left[f_{tf} + \frac{2}{3}(\sigma_x + \sigma_t)\right]$$

$$\tau^2 - \sigma_x \sigma_t \leq 2\frac{f_{tj}}{f_{cj}}\left[0.6 f_{cj} - \sigma_x - \sigma_t\right]\left[f_{tf} + \frac{2}{3}(\sigma_x + \sigma_t)\right]$$

When σ_x<0, the above conditions are replaced by:

$$\tau^2 \leq 0.25 f_{tj}\left[f_{tf} + \frac{2}{3}\sigma_t\right]$$

In the above equations:

σ_x = stress normal to the beam section, calculated from the net section of the beam.

σ_t = normal transverse stress, i.e. that acting on the internal surface parallel to the neutral fibre of the beam and perpendicular to the mean plane of the component, calculated from the net depth b_n.

τ = shear stress in the component, calculated from the net depth b_n.

f_{tj} = tensile capacity of the concrete *j* days after casting.

f_{tf} = flexural capacity of the concrete *j* days after casting.

f_{cj} = compressive capacity of the concrete *j* days after casting.

The capacity values f_{tj}, f_{tf} and f_{cj} are material properties, while σ_x, σ_t and τ should be determined according to the French recommendations [1] and rely on the transverse prestressing that is applied to the section.

For the ultimate limit state, the shear strength V_u is given by:

$$V_u = V_{Rb} + V_a + V_f$$

Where:

V_{Rb} = the term for the participation of the concrete.

V_a = the term for the participation of the reinforcement.

V_f = the term for the participation of the steel fibres.

These terms are to be determined according to the French recommendations [1].

Design theories

In the initial design stage of the UHPC roof, only the linear-elastic part of the stress-strain diagram will be used, so that the structural behaviour of a struture in this material is conservatively approximated. It is noted that the values found in the stress-strain diagram for the ultimate limit state made are more conservative that those determined for the serviceability limit state, when in accordance to the preliminary French recommendations.

This means a linear elastic relation ranging from a tensile stress of 5.6 MPa to a compressive stress of 102 MPa with a modulus of elasticity of 65 GPa will be used in the iterative algorithm. Pre-stressing will be accounted for by moving the origin of the diagram along the curve. Now a larger tensile stress can be obtained, but only by accordingly lowering the compressive capacity. Detailing the pre-stressing, which also includes accounting for creep and shrinkage that lead to prestressing losses, will be done in a later stage of the design.

Durability issues

No special considerations are made for durability issues whilst developing the design. Based on the findings as reported in Paragraph 4.2.4, it is assumed that no durability issues will be of major influence on the structural behaviour of the UHPC. Possible staining of the surface due to corrosion of steel fibres that penetrate the surface and fire protection of pre-stressing cables that are close to the surface are assumed to be avoidable by applying a thin coating that expands to a foam-like material when exposed to fire.

After the preliminary design stage, durability issues will have to be checked based on the French recommendations.

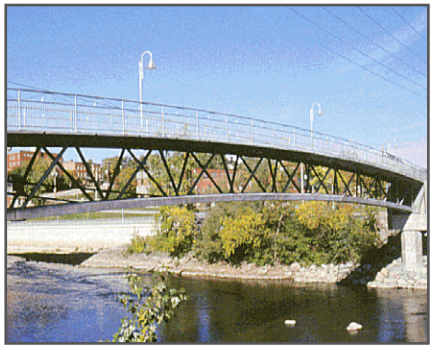


Figure 4.9: Sherbrooke footbridge overview. Picture from Resplendino and Petitjean [48].

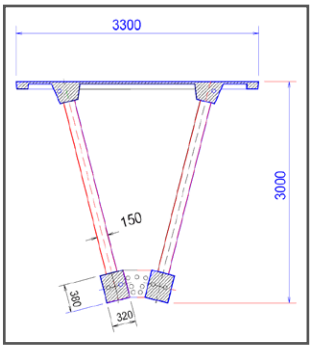


Figure 4.10: Sherbrooke footbridge cross section. From Resplendino and Petitjean [48].

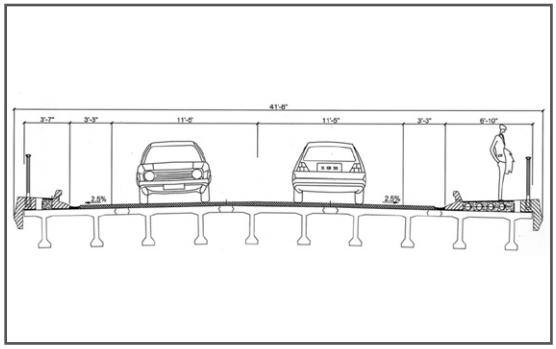


Figure 4.11: Cross section of the Bourg lès Valence traffic bridges. Picture from Hajar et al. [32].

4.5 Examples

4.5.1 Sherbrooke footbridge

Built in 1997, the Sherbrooke pedestrian bridge in Quebec, Canada is the world's first engineering structure designed with UHPC. The pre-cast, pre-stressed pedestrian bridge spans 60 metres, and consists of six pre-fabricated 10 metre elements. Reactive powder concrete was cast in very thin-walled stainless steel tubes [48].

Interestingly, the shape of the bridge is very much based on the truss, which is a conventional shape for steel structures.

4.5.2 Bourg lès Valence traffic bridges

The world's first road bridges in UHPC were built in France in 2000-2001. Two spans of 22 metres are spanned with an assembly of five π-shaped pre-cast beams made from BSI. The concrete is fully self-compacting. If the bridges would have been built in conventional pre-stressed concrete, about 3 times as much concrete would have been needed [31, 32].

The π-shaping is an interesting shape optimisation in the beam design.

4.5.3 Seonyu footbridge

The Seonyu footbridge in South Korea links the main town of Seoul to Sunyudo Island. The main structural element is an arch spanning 120 metres. Pedestrians walk directly on the central part of the arch. The other parts of the arch carry the walkway through a steel structure. The bridge was constructed in 2001-2002 [21].

The arch consists of pre-stressed fibre-reinforced Ductal® segments and has a span over height ratio of 8. Only 120 m^3 of UHPC is used and no conventional reinforcement is applied. The segments are π-shaped; a ribbed deck is supported by two webs. This shape is chosen because it makes the demoulding easier. The bridge is so slender that it is vulnerable to vibrations. Tuned mass dampers are introduced to control the accelerations [21].

The Seonyu footbridge is the largest span in UHPC to date. This structure proves the suitability of the material for this kind of spans. Just like the Bourg lès Valence traffic bridges the designers used π-shaped elements.

4.5.4 Sakata Mirai footbridge

The Sakata Mirai footbridge is constructed in Ductal® UHPC in 2002. It spans 50 metres in one go. The bridge is externally pre-stressed. A total of 22 m^3 of concrete is used in the bridge design, yielding a self weight of 560kN. This is one fourth of the weight that an ordinary pre-stressed concrete bridge for this span would have weighed [57].

The most catching feature of the bridge is the interesting cross section; a box girder with holes in the sides. These holes are employed for the sake of design value and the reduction of self weight [57]. The optimisation process of this bridge led to a very different shape than the π-shapes found earlier.

4.5.5 Shawnessy train station

In the winter of 2003-2004, the roof of the Shawnessy train station was built in the ultra-high performance concrete Ductal® [47]. Twenty-four pre-cast concrete canopies spanning 5 by 6 metres make up the platform roofs. The canopies

Figure 4.12: Overview of the Seonyu footbridge.

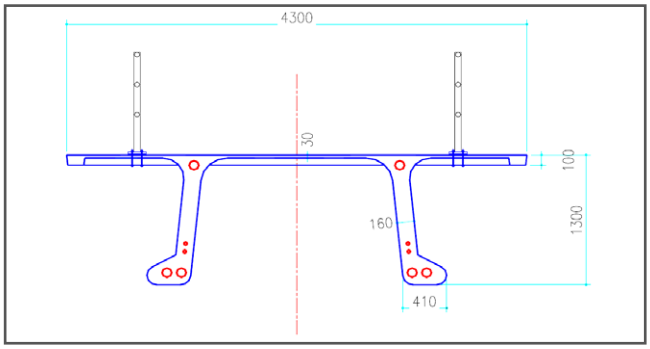


Figure 4.13: Cross section of the Seonyu footbridge. Picture from Resplendino and Petitjean [48].

Figure 4.14: The Sakata Mirai footbridge. Picture from Ten Voorde [65].

Figure 4.15: STS during construction. From Vicenzino et al. [59].

Figure 4.16: Overview of the Millau toll gate. Picture from Structurae [74].

have a thickness of 20 mm only. The important feature of this example of UHPC is the thinness of the shell, which is possible because of the low permeability of the concrete and the fibre reinforcement [59].

4.5.6 Millau viaduct toll-gate

The toll gate of the Millau viaduct in the south of France was built in BSI in 2004. The structure spans 98 by 28 metres. The 1000 m^3 of BSI that is used weighs 2800 tonnes. Forty-nine pre-fabricated elements being 2 metres wide, 28 metres long and 20 to 80 centimetres thick are connected by internal longitudinal pre-stressing [32].

Figure 4.17: Millau toll gate element. UHPC is positioned at the outside. Picture from Servant [53].

CHAPTER 5 DESIGN ALGORITHM

An ant system based algorithm is developed for structural topology optimisation in this thesis. As explained in Paragraph 1.2.2, it is suspected that this particular concept is very well applicable to the problem considered. Moreover, the concept is not widely explored for such applications and academic curiosity is another reason to choose the ant system as a starting point for the algorithm.

This chapter will elaborate on the way the concept is applied to structural topology optimisation, how the concept is implemented, which extensions are proposed to make the algorithm more efficient and finally, how the resulting algorithm works.

5.1 Applying the concept

5.1.1 Objective

As stated in Paragraph 1.3, the objective of this thesis is to create and evaluate a structural design algorithm, based on the concept of the ant system that designs a roof for the Omnisport building in ultra-high performance concrete. The algorithm should be capable of finding an optimal topology of a structure under given boundary conditions. The boundary conditions consist of a space available for the placement of structural elements, the load case or load cases applicable to the structure, the supports and a definition of the structure's properties that together determine its performance.

5.1.2 Representation

The ACO metaheuristic framework can only be applied to discrete problems. Representing the optimisation problem in a discrete way is therefore inevitable.

The space available for the placement of structural elements is made discrete by dividing it into small elements as is shown in Figure 5.1. Each element has two possible states: it can contain material or can be a void. A structure containing i elements that all have two possible states, has 2^i possible solutions.

The determination of the value of a structure is represented in a mathematical way: the *performance function*. This function can be any mathematically definable function.

5.1.3 Optimisation principle

When a search space has been defined, the optimisation principle inspired by ant behaviour can be introduced. As explained in Paragraph 3.1, real ants probabilistically choose paths and lay pheromone on the paths they choose. The pheromone on a shorter and thus better path is reinforced more often and therefore becomes stronger.

In the application of the concept to the considered structural topology design problem, the optimising behaviour is not aimed at finding the shortest path from A to B. Instead, a structure that performs best according to some performance function needs to be found. In the application the ants march the full graph. They go over every element that might

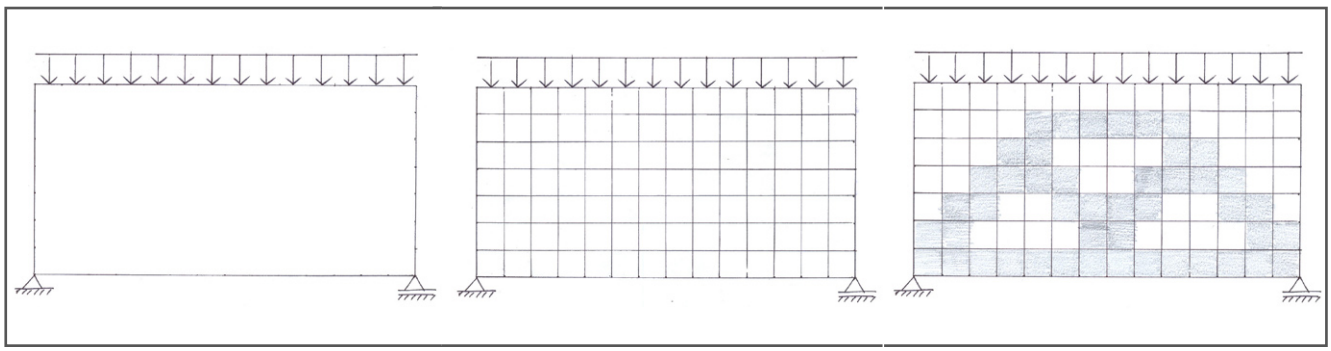


Figure 5.1: Sketch of design space, its discrete counterpart and a possible optimum.

or might not form part of the structure, as will be explained in this paragraph. This discrepancy from real ant behaviour is important to keep in mind whilst reading this paragraph.

Example

The proposed approach – based on the ant system – is explained by an example as shown in Figure 5.2. This is a single-element optimisation problem. The element has two nodes. On the top node (A) it is pinned and on the bottom node (B) a force is applied. The element can either contain material or it can be a void. It is obvious to see that this system will only work if the element actually contains material. This problem forms a good example for explaining the proposed principle.

The choice that has to be made is a discrete problem with two possibilities.

- The element contains material
- The element is a void

This choice can be represented by choosing one out of two paths.

Assume a colony of ants needs to walk from A to B. It can choose to follow path I or path II. For all ants that choose path I, we say those ants choose to put material in the element, while all ants that choose path II choose to leave a void. Now when all ants have walked the path the structures they made can be tested by analysing them in a finite element package.

The structure chosen by the ants that put material in the elements works fine. In other words, its *performance* is good. The system consisting of only a void fails and hence has a bad performance. Now let the ants that chose the well-performing system put pheromone on the path they chose.

In a next iteration, the ants find a different amount of pheromone on path I and II. Ants make a probabilistic decision based on the amount of pheromone they find on a trail. Opposed to the first iteration, more ants now choose to follow path I. After following the path and hence creating a well-performing system, the amount of pheromone is further reinforced. All ants will quickly converge to this one path.

In this system pheromone is assigned to the trails as a function of *performance* rather than path length. This difference from real ant behaviour is the major adjustment that is made to fit the ant system to this optimisation problem.

Extension

The example can be extended by splitting the element into three separate elements. Now the ants need to make three choices in order to construct a system. Again, each path that the ants choose along the way receives an additional amount of pheromone according to the performance of the system. This is sketched in Figure 5.3. The problem has a total of 8 possible solutions, which is split in three decisions of selecting one path over another.

Two dimensions

The two-dimensional problem as sketched in Figure 5.4 is considered. Every element is numbered. Through mapping, a trail is constructed that the ants need to follow. They need to choose one out of two paths for every element, again similar to the choice of assigning material to an element or leaving a void. Again, a number of ants walks the graph, choosing paths based on an initial pheromone distribution, and thus constructing solutions. When all the ants, separately from each other, have completed their trails the structures they have generated accordingly are tested in a finite element package. The pheromone on the paths they chose is updated according to the performance of the structure they constructed.

Now another run can take place in which each of the ants in the colony again constructs a solution choosing paths on the same graph, using the updated amounts of pheromones. The small change in pheromone distribution gives a positive feedback to the system; well-performing solutions are more likely to be chosen again, but leaves room for exploration of other possible solutions and thus reduces the algorithm's sensitivity for getting stuck in local optima.

For this particular problem, the number of possible solutions is 2,097,152, making it impossible to try every single one of them, and the metaheuristic approach starts to pay off.

Three dimensions

The same concept can be extended to three dimensions; again the problem is mapped to choices for paths. Each of these choices represents the choice to either leave a void or put material in an element. The elements together form a structure that has a certain performance according to which the pheromone on the paths is adjusted. This pheromone determines the probability that ants make a certain choice.

Terminology

In order to show the link between the proposed algorithm and the real ant behaviour, some terminology will be used throughout this thesis.

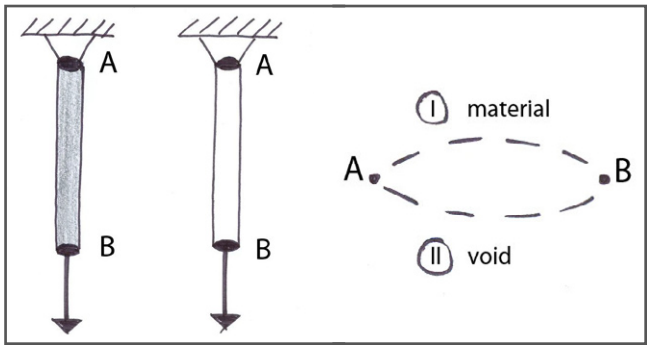


Figure 5.2: Sketch of a single-element structural optimisation problem.

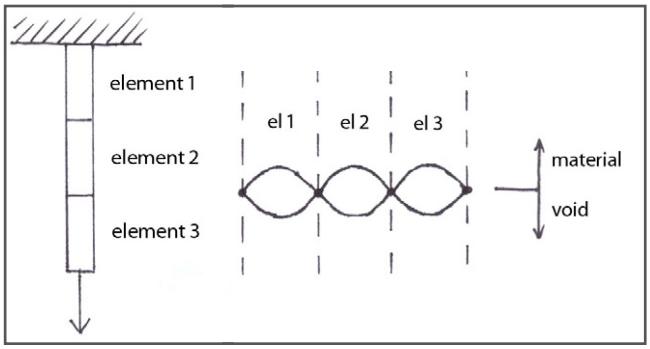


Figure 5.3: Three-element structural problem.

- An *ant* or *agent* is an artificial creature creating one complete possible solution in each iteration.
- A *path* is a connection that an ant can choose as a part of its trail representing the choice to put material or leave a void in an element.
- A *trail* is the combination of paths forming a complete system.
- A *colony* is the group of ants that simultaneously construct trails. The number of ants in a colony is therefore equal to the amount of structures that is generated within one iteration.
- The *pheromone* on a path represents the probability that this path is chosen by an ant as a part of its trail.
- An *iteration* is the process in which the complete colony of ants constructs solutions and these solutions are tested. The pheromone distribution is updated at the and of an iteration. Each iteration uses a different pheromone distribution.

5.2 Implementation

5.2.1 Software and machine

In order to implement the developed concept, a script has been made in Fortran 77 and has been connected to a finite element package. The algorithm is developed around the FEM package DIANA 9.2 [7]. To make the routine as quickly as possible, linear elastic elements are used. Linear elastic calculations in DIANA are relatively simple. In 2D, the plane stress element Q8MEM is used (Figure 5.5). This element has four nodes and needs four integration points. In 3D, the solid element HX24L is used (Figure 5.6). This element has eight nodes and needs eight integration points.

Runs are performed on a Linux machine. The machine has two quad core processors, each core being a 2.66 GHz processor. It has 24 GB RAM and a 1333 MHz front side bus. Each run uses one of the eight cores on the machine. For this reason several runs can be performed simultaneously. If this is done, the runs need to share the internal memory.

5.2.2 Conceptual flowchart

The algorithm is implemented as described in the conceptual flowchart in Figure 5.7. An optimisation run consists of *k* iterations, a colony of *j* ants and a structure of *i* elements.

The process can be split in a few important steps. The process of assigning material or a void to each of the *i* elements is the *structure generation*. After this step the *performance* of the resulting structure is determined. As a last step, the *pheromone update* process is performed. Now a next iteration can start.

5.2.3 Structure generation

In each iteration, each ant generates a structure. The generation of the structure consists of the choice to put material or leave a void in each element. The chance that an ant chooses to put material is identical to the chance that the ant chooses *not* to leave a void, and vice versa. Because of this property, pheromone can be expressed in one value only: the chance that an ant chooses to put material in an element.

In the algorithm the pheromone amount is referred to with the term *reference value*, which is equal to the chance an element is chosen as a part of the solution. This reference value ranges between 0 and 1. If a randomly generated number in the same range is smaller or equal to the reference value, material is assigned to the element and if the number is larger, a void is left.

Randomness

The generation of a structure is based on randomness. This randomness means that the structures that are explored during the process will be different for every run. Through pheromone reinforcement these structures influence the direction in which the algorithm starts its search. Due to this randomness, different runs of the same algorithm on the same problem will lead to slightly different results. Also, it is possible that no good solutions will be found at all. Or that, after a few iterations, the run stagnates because no sound solutions are found anymore. It is therefore advisable to perform several parallel runs when using the algorithm. This way, the influence of the randomness becomes clear and the *random fluctuations* can be filtered out. Interestingly, Goss et al. [30] showed that the same random fluctuations occur when real ants choose their path over a double bridge.

At the same time the random number that forms the basis of the probabilistic choice to leave material or a void in an element is not entirely random. The package DIANA offers a function that generates a series of numbers that are distributed normally in between 0 and 1 [7]. This series needs a *seed* to be able to start. For the second number that is generated,

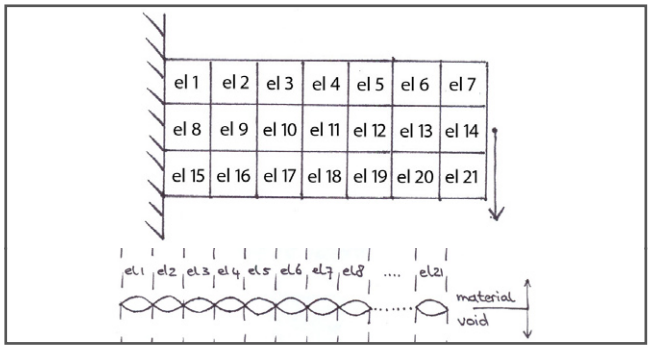


Figure 5.4: Two-dimensional structural optimisation problem and paths choice.

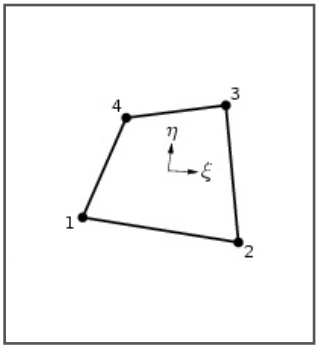


Figure 5.5: DIANA Element Q8MEM. From De Witte [7].

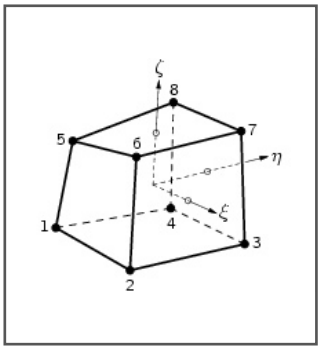


Figure 5.6: DIANA Element HX24L. Picture from De Witte [7].

the first generated number will function as a seed, et cetera. Because of this, random strings will be identical when the same seed is used. One has to keep in mind that this process of generating a number in the range of 0 to 1 can be a weak link in the process.

In the algorithm the number of seconds on the computer's clock at the start of the process helps generating the random seed for each structure that is generated. It is assumed that the user will not control this parameter. However, there is a chance of 1/60 that this number is similar for two runs. In that case, the runs will yield exactly the same results. The random seed used at the start of the generation of a new structure is:

$$S = j \cdot (t_1 + 1) + k \cdot (t_2 + 1)$$

Where:

S = random seed.
j = number of the current structure.
t_1 = first digit of the number of seconds at the start of the run.
k = number of the current iteration.
t_2 = second digit of the number of seconds at the start of the run.

The number one is summed with the digits obtained from clock of the system so that the situation that zero is multiplied with either the number of the structure or the number of the iteration is avoided. This situation would mean all the structures within the same iteration, or all the structures with the same number over the different iterations are generated from the same seed.

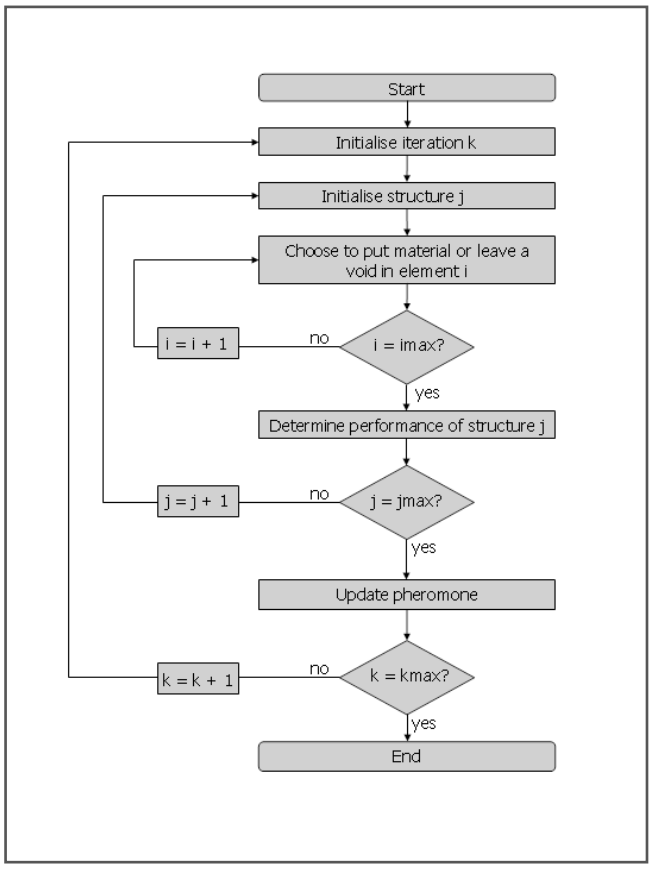


Figure 5.7: Conceptual flowchart of the program.

As an example, the random seed for the 6th structure of the 12th iteration of a run that started at 11:23:59 is $S = 6*(5+1)+12*(9+1) = 156$. This example also shows that the seed for the 11th structure of the 9th iteration is the same, which means the seed is not unique for every structure.

Void

It is not possible to leave an element empty in a finite element calculation. The void that the ant is supposed to leave in an element when it chooses to do so is therefore approximated with a very flexible material. In the algorithm this material is assigned a modulus of elasticity of 1 Pa and a Young's modulus of 0. The material is supposed to act like a sponge. It will not attract stresses when placed in between structural elements. The structure will show big displacements when two structural elements are connected by such a sponge-like material. When a force is applied on the sponge-like material, it will have to be able to drape over the structural elements and transfer the applied forces to the elements that do contain material. The functioning of this sponge-like material will be tested in Paragraph 6.1.3.

5.2.4 Performance determination

Structures are valued according to a performance function. The performance function can contain many variables and can be different for one, two and three dimensional structures. In this paragraph the performance function variables as developed during this thesis are described. The algorithm is developed for two-dimensional optimal structural topologies in ultra-high performance concrete.

Stresses and displacements

Stresses and displacements should stay within certain boundaries for the generated structures. The structural properties of the generated solution are determined based on the results of a linear structural analysis in DIANA. In this thesis, only the stresses and displacements are used from this analysis. It is possible to use multiple load cases.

Volume

In general it is more economical to use less concrete in the structure. Therefore the volume of each structure is calculated.

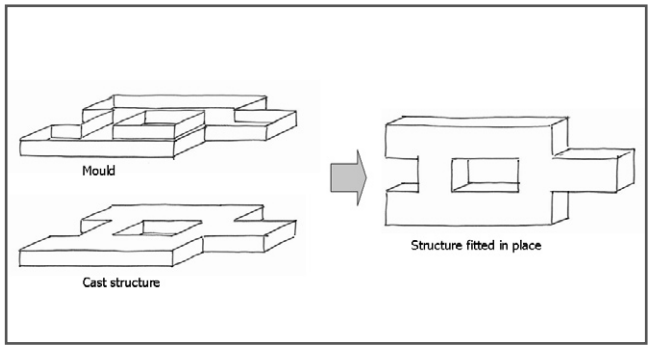


Figure 5.8: Concrete casting and mould surface.

Mould surface

In order to make the concrete elements, a mould has to be made. Even though the complexity of the shape of the mould is very important, the surface of the mould can be used as a parameter to estimate the cost of the mould.

The location of the elements relative to each other depends on the mesh that is used and is known for each problem. As will be explained in Paragraph 5.2.6, the algorithm can look up which are the neighbours of an element. It is also known to which elements material is assigned. Because of this, the algorithm can measure the surface area of the concrete that has to be moulded.

The mould covers the areas in between the elements as well as the back of the structure. It is assumed that the concrete structure is cast laying flat on the ground. This and the mould area that follows from this are shown in Figure 5.8.

Pre-stress level

Pre-stress is often applied to UHPC in order to increase its tensile capacity. However, pre-stressing brings along extra costs. Structures with a high level of pre-stress are therefore more expensive.

The algorithm calculates which level of pre-stress is needed to be able to cope with the tensile stresses. The level of pre-stress for an element is:

$$VP_i = \frac{\sigma_{max;occur} - f_t}{f_{y;max}}$$

Where:

VP_i = Pre-stress level of an element
$\sigma_{max;occur}$ = Maximal occurring stress in the element
f_t = Elastic tensile strength of the concrete
$f_{y;max}$ = Maximal tensile stress for fully pre-stressed element

The pre-stress level of an element varies in the range of 0 to 1.

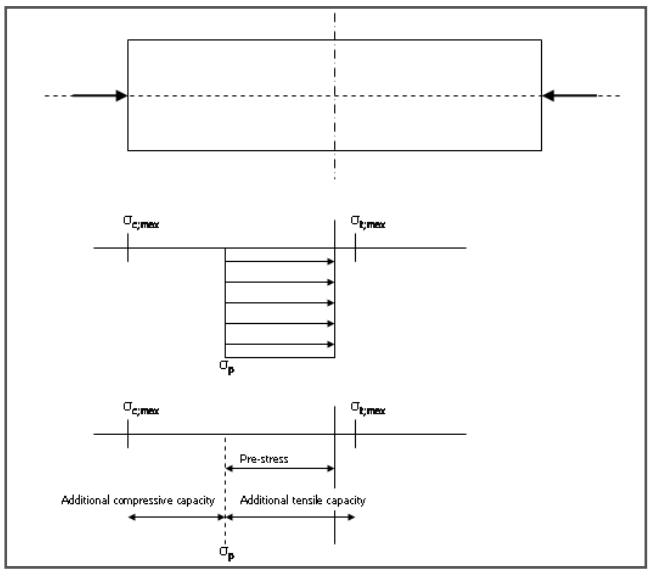


Figure 5.9: Pre-stress lowers the compressive capacity.

The pre-stress level of a structure is determined according to the following function.

$$VP_{tot} = \sum_{i=1}^{imax} VP_i \cdot V_i$$

Where:

VP_{tot} = Pre-stress level of the structure
V_i = Volume of element *i*

This pre-stress level of the structure is a measure for how much pre-stressing should be placed in the structure such that is can handle the tensile stresses. Three difficulties exist with this method.

Firstly, applying a pre-stress to a concrete element means the element is loaded under compression. This lowers the possibilities of applying an additional compressive stress to the element. In Figure 5.9 an element is considered that is pre-stressed. The pre-stress loads the element to a certain level. Now the tensile capacity additional to this pre-stress is larger than the original tensile capacity, but the additional compressive capacity is lower than the original compressive capacity. Applying a pre-stress in order to be able to cope with the tensile stress in fact means that the elastic range is shifted. Now there is a possibility that the compressive stress cannot be coped with even though its value is lower than the compressive strength. This phenomenon is coped with within the proposed algorithm by lowering the allowable compressive strength according to the pre-stress level that is applied.

The second difficulty is the orientation of the pre-stress in different adjacent elements. Pre-stress is applied in strands that are preferably as continuous as possible. Curves in the orientation of these strands are problematic, not only because these curves are very difficult to manufacture, but also because a sideways reaction force within the concrete is generated that causes an extra load to the concrete. Within the proposed method, the fabrication of pre-stress and the generated sideways loads are not considered. The results should be analysed to see if these phenomena cause any problems. Figure 5.10 visually shows how the pre-stressing strands could be applied and how discontinuous pre-stress could cause problems.

Lastly, tensile stresses can occur in any directions, which is a difficult thing to implement in the algorithm. If the applied loads are either in the same direction or in the exact opposite direction, each element will have only two principal directions as is shown in Figure 5.11. In the proposed algorithm

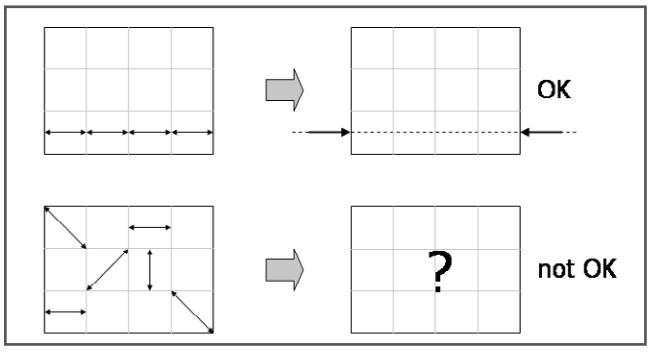


Figure 5.10: Continuous and discontinuous pre-stress.

such load cases are the only load cases considered. In such cases, pre-stressing might be needed in two perpendicular directions. The algorithm is developed such that it can cope with this situation.

Holes analysis

A hole in the structure is defined as a group of voids that together have a certain size and has material on all four sides of it. Holes like these can be traced by using the knowledge about the position of the elements relative to each other. A more in-depth description of this process is given in Paragraph 5.3.3. Holes can be used to place ducts and walking bridges for maintenance, for example. These elements do not have a strict location but need to be accounted for in the design. In the hole analysis the routine counts how many holes exist in the structure that have the minimum size.

Performance function

The structure's performance is determined with a mathematical function. This means any mathematically definable performance can be optimised for. In the proposed algorithm, all the above specified analysis criteria are used as a performance criterion at some point. Anything could be a performance criterion, however, as long as it is measurable in the algorithm.

The performance function as used in most of the runs that are performed during this thesis is the following.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable} \\ \dfrac{1}{C_V V + C_S S + C_P P V_P + C_H H} & else \end{cases}$$

In which:

V = Volume of the generated structure (m³).
S = Surface of the mould (m²).
P = Pre-stress level of a certain element (-).
V_P = Volume of a pre-stressed element (m³).
H = 0 if the holes criterion is fulfilled and 1 otherwise.
C_V, C_S, C_P and C_H are constants.

The above equation determines the performance of a structure. If the generated structure does not meet the boundary conditions in terms of stress and displacement the performance is set to a very low number. The reason the performance is not set to exactly zero, is that now the user can see in the output files on which criterion the structure failed. Structures that failed on stress capacity get assigned a slightly different performance than those that failed on maximum displacement. The assigned performance is still about zero, but the reason of failure can now be seen in the output files.

If the boundary conditions are met, the performance is set to be the inverse of the estimated costs. Material volume, mould surface and amount of pre-stress together can give an estimation of the costs of the structure. The constants that are part of this formula will be determined later on. An extra parameter is the holes criterion. The structure is penalised for not having enough holes to provide space for ducts and walking bridges by adding a virtual cost to the structure's estimated costs.

It is noted that the performance function is a parameter that the user can set when a run is started, and is very specific to the problem.

Additional performance parameters

Anything measurable in the algorithm can be used as a performance parameter. In 3D, where a whole building can be topologically designed, one can think of considering room sizes and routing problems.

5.2.5 Pheromone update

Element performance

In the ant system as proposed by Dorigo et al. [27], the performance of a path is calculated as the sum of the performances of the trails it was part of. For the structural cases considered here this means that the performance of an element is the sum of the performances of the structures that were generated in an iteration in which material was assigned to this element. This would be done by the routine in the following manner.

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material \\ perf_{i,k} & if\ i = void \end{cases}$$

With:

$perf_{i,k}$ = performance of element i in iteration k
$perf_{j,k}$ = performance of solution j in iteration k

In addition to the performance of a structure, an extra parameter determines the performance of an element. A *daemon action*, a centralised action that cannot be performed

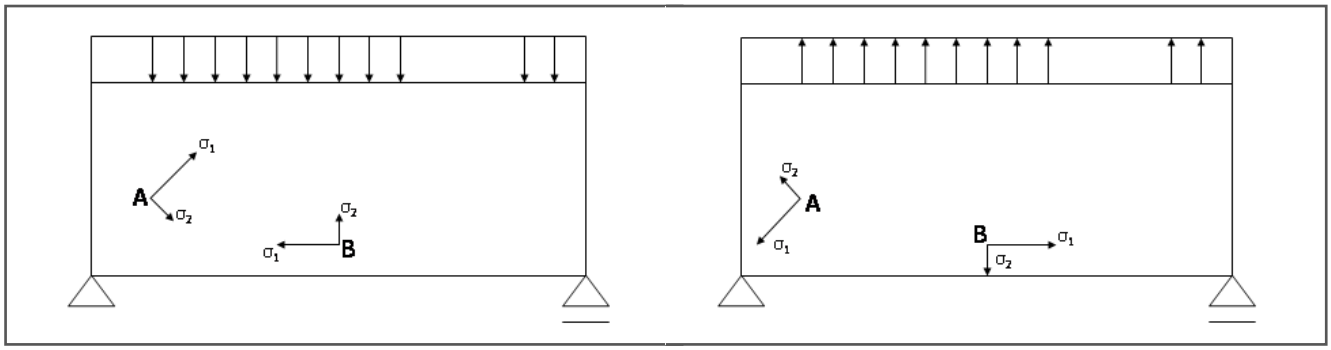


Figure 5.11: Principal stresses due to two load cases in opposite directions. The principle stresses are each others opposites.

by a single ant [8] limits the performance of an element that was part of a well-performing structure, but had a relatively small stress. This extension to the conceptual algorithm will be discussed in Paragraph 5.3.1.

Element pheromone

Now when structures are generated and analysed and the performance of the structures and thereafter the elements is calculated, the pheromone on the paths that the ants can choose is updated. In the proposed algorithm, pheromone is expressed in a reference value for each element. At the end of an iteration the reference value is updated according to the relative position of element performance in comparison with the performance of the best performing element, according to the following formula.

$$RV_i(k+1) = (1-\rho)RV_i(k) + \rho \frac{perf_{i,k}}{perf_{\max,k}}$$

With:

$RV_i(k+1)$ = reference value of element i in iteration $k+1$
ρ = pheromone decay factor
$RV_i(k)$ = reference value of element i in iteration k
$perf_{i,k}$ = performance of element i in iteration k
$perf_{max,k}$ = largest element performance in this iteration
$perf_{j,k}$ = performance of solution j in iteration k

In this function the factor ρ makes sure pheromone evaporates over time. A larger pheromone decay factor makes the algorithm more greedy; newly developed well-performing solutions highly affect the pheromone distribution. Optima will be approached quicker, but the algorithm is more sensitive to getting stuck in a local optimum.

The pheromone update process is one of the most important processes in ACO algorithms [8].

5.2.6 Data handling

The data needed for the algorithm is stored in a few matrices and arrays.

DATSTR matrix

The mesh, the discrete counterpart of the space available for placing structural elements, consists of i elements. Each element is numbered. An artificial ant makes a probabilistic decision to place material in an element or leave a void in each element. The structure that is constructed in this way is stored as an array consisting of i entries, each being one or zero. A one means that for material is placed the element associated with this array entry for this particular structure. A zero means the element is left empty. This representation is depicted for the possible optimum solution in Figure 5.12. Besides the structure itself, some properties of the structure need to be stored. Maximum displacement, volume, minimum and maximum stress in the structure, performance and the rank within the iteration are stored at the end of the array that defines the structure.

Each iteration consists of j ants, that each constructs a solution or structure. All the arrays of the structures generated in an iteration form a matrix of j rows. As a $(j+1)^{th}$ row, the best solution found so far is stored. As a $(j+2)^{th}$ row, the pheromone of each element is stored. The last, $(j+3)^{th}$ row, is used for the performance of each element as determined in that iteration. This matrix is remade during each iteration. This data storage matrix DATSTR((i+8),(j+3)) is the most important data structure in the algorithm. The DATSTR matrix is visually represented in Figure 5.13.

Stress matrix

Parallel to this DATSTR matrix, a three dimensional stress matrix STMIMA(i,(j+1),3) stores for each element in each structure the maximum principal stress, the minimum principal stress and the resulting Von Mises stress, resulting in 3 matrices of i columns and $(j+1)$ rows. The one extra row is for the best solution found so far. The stresses in this matrix are used in the determination of the performance of the elements.

Element position matrix

A last data handling tool is an elements position matrix. During the optimisation process, the mesh is kept constant. The position of the elements relative to each other is implemented in the routine by creating a matrix where all the element numbers are stored at their proper position. The matrix has two or three dimensions, depending on the number of dimensions of the problem considered.

5.3 Extensions

The algorithm as described above is capable of selecting elements for a structure probabilistically and assigning material to those selected. It can perform a linear elastic analysis and can calculate a performance based on stresses, displacements, volume, mould surface, pre-stress and holes that are available for functions.

The process is already quite extensive. It has performance variables that are developed for a design case for two dimensional ultra-high performance concrete structures. The optimisation process lies in the finding of new solutions with a better performance. In order to make this process more robust, efficient and intelligent, a few extensions have been developed.

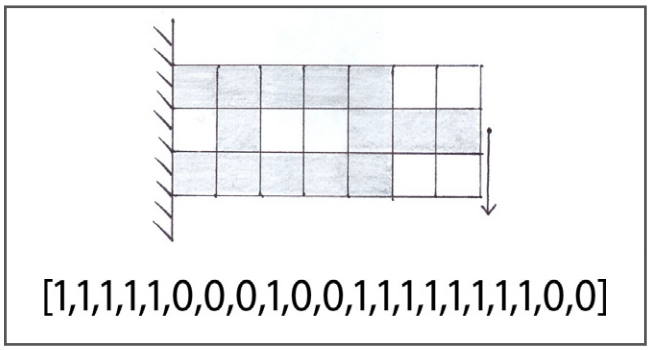


Figure 5.12: Possible solution of the two-dimensional problem and its array representation.

5.3.1 Stress daemon

As explained in Paragraph 5.2.5, the performance of elements is determined in the following manner.

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material \\ perf_{i,k} & if\ i = void \end{cases}$$

With:

$perf_{i,k}$ = performance of element i in iteration k
$perf_{j,k}$ = performance of solution j in iteration k

However, in a structure, some elements are stressed to a higher level than others. It is proposed to add an extension that limits the performance of elements that were part of a well-performing structure, but had a very low stress, to a certain percentage of the structure's performance. Because the pheromone update process is based on the performance of an element, this *daemon action* will limit the pheromone on this element. In other words, the element will have a lower probability of getting chosen again. This stress daemon therefore functions like a rejection list. This extension makes sure that more problem-specific knowledge is incorporated in the algorithm, and possible makes the algorithm more efficient.

It is noted that the optimisation process in the proposed algorithm is multi-objective. It can be that an element has a very low stress, but that is fills up a small gap and therefore reduces the moulding surface, or that it connects parts of the structure which would otherwise collapse. It can, in other words, still be essential for the high performance of the structure. A stress daemon as proposed that acts like rejection nomination therefore suits the considered optimisation of structures better than just deleting such an element, as is done in ESO [19]. The user is given the option to refine the stress daemon into two steps. The performance reduction can be set differently for small stresses and very small stresses in this manner.

The performance of an element is now calculated as follows.

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ \sigma_i \geq \sigma_{threshold} \\ perf_{i,k} + c_{deamon;2} \cdot perf_{j,k} & if\ i = material\ and\ \sigma_{threshold;1} \leq \sigma_i < \sigma_{threshold;2} \\ perf_{i,k} + c_{deamon;1} \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < \sigma_{threshold;1} \\ perf_{i,k} & if\ i = void \end{cases}$$

With:

$perf_{i,k}$ = performance of element i in iteration k
$perf_{j,k}$ = performance of solution j in iteration k
$c_{daemon;1}$ = the daemon constant for the smallest threshold
$c_{daemon;2}$ = the daemon constant for the second threshold
$\sigma_{threshold;1}$ = the smallest stress threshold
$\sigma_{threshold;2}$ = the second stress threshold

If material was assigned to the element in a structure, the performance of that structure is added to the performance of the element, unless the stress in the element was below the daemon threshold. In that case the performance of the structure is multiplied with a daemon constant $(0<c_{daemon}<1)$ before being added to the performance of the element.

5.3.2 Connectivity analysis

In Paragraph 5.2.6 it was explained that the position of the elements is stored in a matrix. The connectivity analysis consists of a routine that goes through this matrix. If an element is assigned material, it checks whether its neighbours are assigned material as well. If the element is on the edge of the mesh, one neighbour will be enough for the element to be considered connected, but if the element is in the middle of the mesh it needs two neighbours. If the element does not have enough neighbours, the material that was assigned to it is replaced by a void.

The connectivity analysis is not 100% bug-free. If there happens to be a group of elements that is not connected to the rest of the structure, the elements are connected to others but the group as a whole is not. The connectivity analysis will not remove such a group. However, the chance that a group of unconnected elements emerges is much smaller than the chance that one or two unconnected elements emerge. The connectivity analysis as implemented suits its basic purpose, but leaves room for improvement.

5.3.3 Holes

The holes analysis has been described earlier as a performance parameter. The determination of this parameter is quite extensive and not trivial in this kind of optimisation routine. Therefore it is further explained in this paragraph. The holes determination is based on a boolean matrix. The matrix represents the mesh that is used in the routine and has all false entries, except for those entries that correspond to elements that are nominated to be part of a hole. These *nominees* are voids that are surrounded by elements to which material is assigned. Holes in the structure are by definition surrounded by material, which is the reason for this nomination criterion.

		ELEMENT 1	ELEMENT 2	ELEMENT 3	ELEMENT 4	ELEMENT 5	ELEMENT 6	ELEMENT 7	...	ELEMENT I	VOLUME	DISP(SLS1)	DISP(SLS2)	STRESS_MAX(ULS1)	STRESS_MIN(ULS1)	STRESS_MAX(ULS2)	STRESS_MIN(ULS2)	MOULD SUFACE AREA	PRODUCT OF PRESTRESS	HOLES CRITERION MATCH	PERFORMANCE(J)	RANKING
		1	2	3	4	5	6	7		NELS	NELS+1	NELS+2	NELS+3	NELS+4	NELS+5	NELS+6	NELS+7	NELS+8	NELS+9	NELS+10	NELS+11	NELS+12
SOLUTION 1	1	INT	INT	INT	INT	INT	INT	INT	...	INT	INT	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	INT
SOLUTION 2	2	INT	INT	INT	INT	INT	INT	INT	...	INT	INT	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	INT
SOLUTION 3	3	INT	INT	INT	INT	INT	INT	INT	...	INT	INT	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	INT
SOLUTION 4	4	INT	INT	INT	INT	INT	INT	INT	...	INT	INT	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	INT
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
SOLUTION J	NSLS	INT	INT	INT	INT	INT	INT	INT	...	INT	INT	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	INT
BEST SOLUTION UNTIL K	NSLS+1	INT	INT	INT	INT	INT	INT	INT	...	INT	INT	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL	INT
PHEROMONE I (K-1)	NSLS+2	REAL	REAL	REAL	REAL	REAL	REAL	REAL	...	REAL												
ELEMENT PERFORMANCE(K)	NSLS+3	REAL	REAL	REAL	REAL	REAL	REAL	REAL	...	REAL											REAL	

Figure 5.13: Data storage matrix.

The routine goes through all these nominations and checks if the nominated elements have neighbours that are also nominated. If a group of nominated elements has a certain size, it is called a hole and counted by the routine. Then all the entries that were part of the hole change from *true* to *false*, so that they will not be counted as part of another hole.

The number of holes is compared to the number of holes needed. An *inverse matching ratio* is calculated, which is zero if the number of required holes is fully matched and one if there are no holes. This forms the parameter that determines the extra virtual cost for structures that do not have enough holes and is used as described in Paragraph 5.2.4.

5.3.4 Reference value limits

Especially for structures with many elements, it might be interesting to limit the maximum and minimum reference value. By doing so, elements will never obtain a 100% chance of getting chosen, nor do they obtain a 100% chance of being left out. This will stimulate the routine to keep looking for new options and keeps the routine from getting stuck in local optima. Limiting the pheromone explicitly to an upper and lower boundary is an important part of the *MIN-MAX* Ant System as introduced by Stützle and Hoos [55].

A drawback of limiting the reference value is explained with an example. If the reference value has a minimum value of 0.1, which equals 10%, about 10% of the badly performing elements will always be chosen as a part of a structure. This means an optimal structure will be approximated, but never really found. To counteract this problem, the minimum and maximum reference value can be reset to 0 and 1 after a certain number of iterations.

5.3.5 Elitism and solution copying

Letting the best solution have a larger influence in the pheromone update process is called *elitism* and strongly guides the search; the solutions converge quicker to an optimum. As a result, an optimum is approached in less iterations. A drawback is the diminishing of the exploratory behaviour of the algorithm [8]. In the proposed algorithm, a rank-based form of elitism is implemented. Only the best few solutions take part in the determination of the element performance. Additionally, the best performing solution is directly copied into the next iteration.

5.3.6 Multiple load cases

In the algorithm multiple load cases can be considered. The algorithm is developed with the aim of analysing the Omnisport hall roof. During the design process performed by the engineers on the project, only upward and downward load cases proved governing. For this reason, up- and downward load cases are the only cases that can be implemented in the proposed algorithm. Calculations are only performed once per structure; displacements and stresses are determined for a virtual load of 1 kN/m^2. Due to linear elasticity, stresses and displacements for the load cases that need to be considered can be calculated from these results.

It is wise to estimate the self weight of the structure and not calculate the self weight of every structure that is generated. This process would slow the algorithm down. Moreover, the structures generated in the beginning of the optimisation process contain a lot of elements that will get rejected later on. Their self weight might lead to a funny loading pattern and illogical deflections that will make the process more sensitive for getting stuck in local optima.

5.3.7 Three dimensions

The concept can be applied to any problem; the number of dimensions does not have an influence in this. In three dimensional problems, it will be possible to take the size of rooms into account and their position relative to each other. This way functionality of spaces and routing can be optimised for. All this has not been developed as a part of this thesis, but is possible within the same concept.

5.4 Algorithm

5.4.1 Flowchart

The flowchart for the proposed algorithm is given in Figure 5.14. In the flowchart the structure generation, performance determination and pheromone update have visually been specified. The processes with a lighter grey background have significant sub-processes that are not specified within the flowchart. At this point, the whole proposed algorithm will be gone through step by step.

At the start, the parameters are set and the data storage matrices are initialised. Job information is printed to a readable file. The first iteration k and the first structure j are initialised. A random number is generated for each element i. If the number is higher than the reference value, a void is left, otherwise material is assigned.

Now the analysis of the structure starts. First, the connectivity analysis removes unconnected elements. Then the moulding surface and the matching of the number of holes are calculated. A linear elastic analysis in DIANA is performed. Based on the results, maximum and minimum stresses as well as displacements are stored and pre-stress is calculated. Based on the results of all this, the performance of the structure is calculated.

When all the structures of the iteration are generated and analysed, the best few performing structures are traced. The performance of the elements is calculated based on the performance of these *elitist structures* and the stress daemon. For each element, the old reference value is multiplied with the pheromone evaporation factor and new pheromone is added according to its performance. The best solution is copied into the next iteration. Data is printed to a readable file and a new iteration is initialised.

The source code of the algorithm is given in Appendix C.

5.4.2 Parameters

In this paragraph the most important parameters in the proposed algorithm are quickly discussed. A sensitivity analysis for the parameters will be performed in the next chapter.

This paragraph merely gives an overview of the different influences the parameters have on the optimisation process.

Pheromone decay factor

The pheromone deposited by the ants evaporates over time. In the algorithm this evaporation or *decay* of the pheromone is determined by the pheromone decay factor ρ. This parameter determines the influence of structures from the last iteration on the pheromone distribution. A larger ρ lets the old pheromone evaporate quicker and approach newly found optima sooner.

Number of ants

The number of ants j is equal to the number of solutions that are simultaneously constructed within an iteration. The run time of the routine is linearly dependent on the number of ants used.

Number of iterations

The routine needs a certain amount of iterations before an optimum is approached and the solutions have converged. Once a solution is completely converged, no more iterations are necessary.

Size of elite

The size of the elite determines how many solutions take part in the pheromone update process at the end of an iteration.

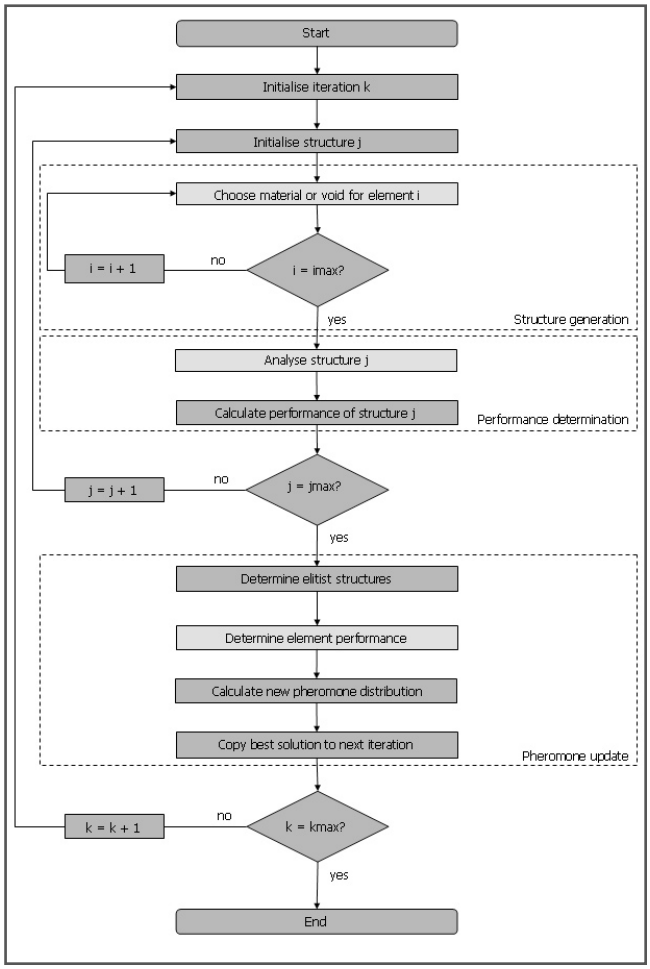


Figure 5.14: Flowchart.

Initial reference value

For the first iteration, the chance that elements get chosen is set by the user. For the structural problems as considered with the proposed algorithm it is wise to take a fairly high initial reference value. This way, many elements will get selected which yields a big chance of finding structures that are stiff and strong enough to cope with the forces applied to them. With these sound solutions as a starting point, the optimisation process can start. Test runs have shown, however, that sound structures can evolve after a few iterations even when there were no sound structures generated in the first few iterations.

Number of elements

The number of elements in the structure that is optimised with the proposed algorithm has a twofold effect on the run time.

In the first place, the number of possible solutions depends on the number of elements in the mesh. A structure containing i elements that all have two possible states, has 2^i possible solutions. This means the search space is bigger and it is more difficult to find an optimum solution in this big search space.

Besides this effect the number of elements also determines how long an ant run takes. More elements mean the process of generating and analysing a single structure become longer.

2D/3D

The algorithm can be applied to 2D as well as 3D problems. Run times go up in 3D because more elements are needed and because 3D elements need eight integration points whereas 2D elements only need four.

CHAPTER 6 VALIDATION AND SENSITIVITY

6.1 Validation

The proposed algorithm is validated on the problem depicted in Figure 6.1. This is a well-explored validation problem in optimal structural design [3]. Michell derived the analytical solution for this problem in 1904 [15]. He showed that the least weight structure for this problem is the structure shown in Figure 6.2. BESO optimises towards minimum weight and its best approximation is the structure shown in Figure 6.3. The similarity is evident.

To validate the algorithm, the problem as given in Figure 6.1 is inputted in the algorithm. A plate, supported by two pins, is loaded in the middle by a downward point load. In the area shown in the figure material can be placed. The performance function is adjusted such that the algorithm looks for the lightest structure within certain boundary conditions for displacements and stresses.

The problem that is analysed is symmetrical. Only half the problem is analysed in the algorithm. This results in a quicker optimisation process. Figure 6.4 shows the boundary conditions as used in the optimisation process. All the results shown are mirrored along the line of symmetry.

6.1.1 Coarse Michell truss

Several test runs have been performed on a coarse representation of the Michell truss validation problem. Runs are presented in Appendix D.1.1. The parameters and boundary conditions of the several runs slightly differ. Because the runs have taken place as a validation early in the development process of the algorithm, not all the features that are part of the final algorithm have been used during these validations.

Input

The boundary conditions and the mesh are set up as shown in Figure 6.4 and Figure 6.5 for half the problem; the structure is cut along the line of symmetry. The structure is pinned in the bottom left corner; translations are restricted, rotations allowed. Along the line of symmetry, horizontal translations are restricted. Half the point load is attached to the structure.

The plate has a thickness of 0.10 metres. The analysed space (half the full space, 2.40 by 1.60 metres) is meshed to a 24 by 16 element mesh consisting of 384 elements. The number of possible solutions is $2^{384} = 3.94*10^{115}$.

The material has the Young's modulus and Poisson ratio of UHPC. As a yield strength, 50 MPa is assumed. This value is not based on the properties of UHPC and is identical for compression as well as tension. The maximal displacement needs to be smaller than 0.15 metres, or span/320.

Multiple runs are performed on this problem and can all be found in Appendix D.1.1. Runs 6 till 10 are done for a point load of 400 kN. Elements get penalised if their stress is much higher than the yield stress. Run 11 performs more iterations with the same boundary conditions. In run 12, the penalty for a stress higher than the yield stress is replaced

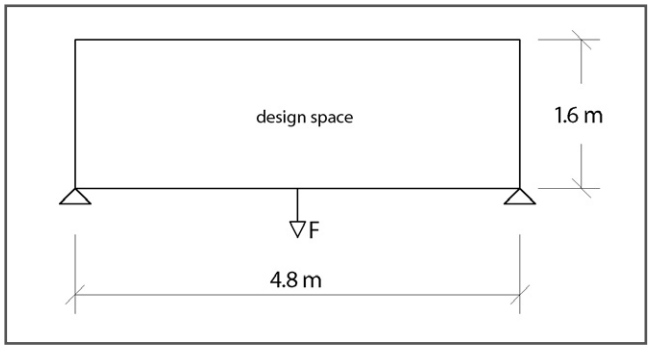


Figure 6.1: Validation problem.

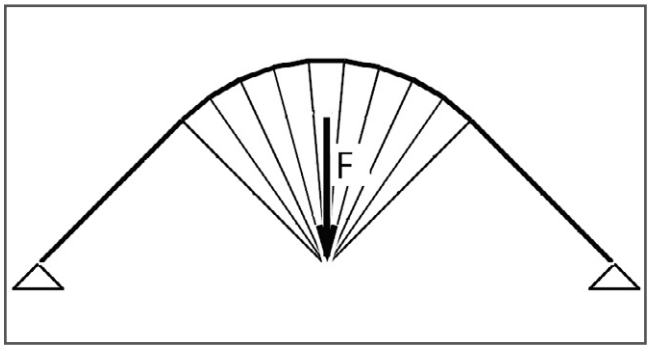


Figure 6.2: Analytical solution to validation problem. Picture based on Kepler [38].

by a boundary condition; structures in which a stress higher than the yield stress occurs are assigned a performance of almost zero, a more strict approach. Lastly, in run 16, the load is increased to 600 kN.

Optimisation process

Specific run data for the different runs is given along with the results in Appendix D.1.1. In comparison to the proposed algorithm as described in Chapter 5, some parts of the algorithm were not in full use during this validation.

The algorithm used an elitist strategy with solution copying. Stress daemon actions were used, but a connectivity analysis and holes analysis were not implemented. The performance function was limited to the following in case of runs 6 till 11:

$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

The last formula shows that the stress daemon was used to penalise elements that were overstressed. The reason this was done is merely because during development, it was interesting to see if the algorithm would work with one strict boundary condition for displacement. When this proved to function properly, a more strict boundary condition for stresses was implemented as can be seen below. The choice to penalise from $1.5 * f_y$ onwards is a rather arbitrary choice made because of the same reason; starting with fairly soft penalties is a good way to see if the concept works. Penalties can be set more strict and tested again once sound solutions emerge.

To the case of runs 12 and 16 the following performance determination applies:

$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-8} & if\ |\sigma|_{max} > f_y \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 0.1 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

Here one can see that a more strict boundary condition for overstressing is implemented.

The performance chart visually represents the optimisation process. Those of runs 6 till 10 are given in Figure 6.6. For the other runs, performance charts are given in Appendix D.1.1. The graphs show the performance of the best solution found up to a certain iteration. The performance grows over the iterations, which means that the best solutions found in the process contain less and less material. In between iteration 15 and 30, the algorithm gets stuck in a local optimum in some of the runs. Further optimisation after iteration 30 shows that the algorithm is capable of getting out of this local optimum. The process is stopped after 50 iterations, but the optimisation algorithm still finds better solutions.

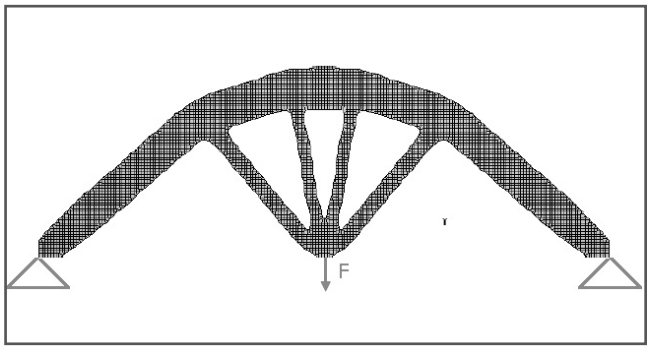


Figure 6.3: BESO solution to validation problem. Picture based on Huang et al. [36].

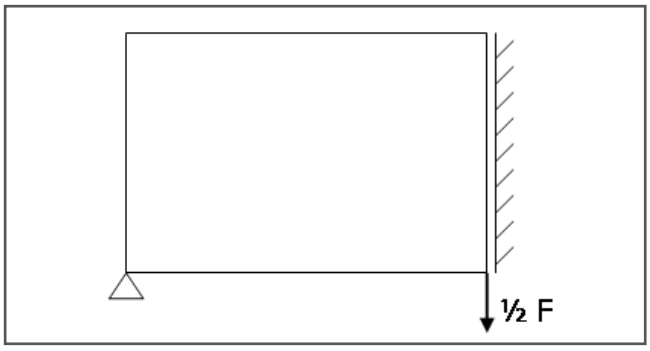


Figure 6.4: Boundary conditions as used in the optimisation process.

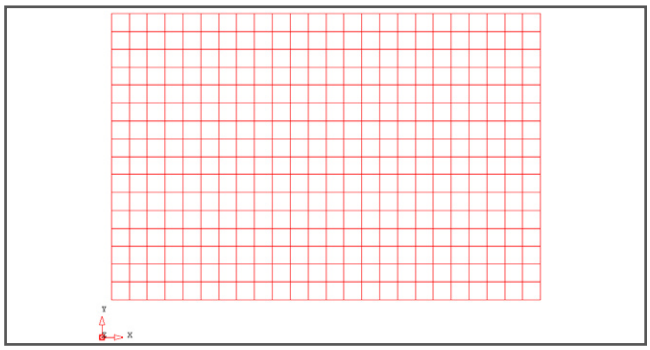


Figure 6.5: Mesh as used in the coarse validation runs.

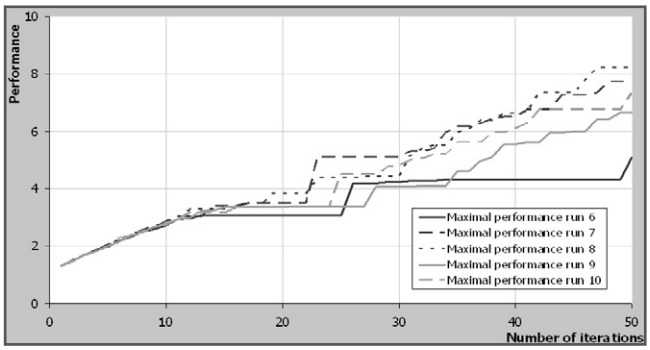


Figure 6.6: Performance chart of five coarse validation runs.

Because the process is not yet completely stagnated after 50 iterations, run 11 performs 250 iterations on the exact same problem. As can be seen in Appendix D.1.1, the best solution is found around iteration 80.

Run 12 uses more strict boundary conditions. This seems to have no effect on the results or the optimisation process. Run 16 also uses the more strict boundary conditions, and is loaded with 600 instead of 400 kN. The performance chart in Appendix D.1.1 shows the performance grows over the iterations without premature stagnation. At the end of the run at 50 iterations, the performance is still growing.

The performance charts of the runs show that the algorithm optimises towards an optimum. Whether the optimum is the global optimum will be analysed in the next sub-paragraph. Most or the runs are stopped before the optimisation process was entirely completed. The system is able to find its way out of local optima. All these observations mean the optimising behaviour is satisfying at this stage.

Output

The algorithm outputs structures that are supposed to be some kind of optimum in terms of light weight structures. In order to check whether the optima found by the algorithm are plausible, they are visually compared to the analytical solution found by Michell.

Due to random fluctuations during the process, each run yields a slightly different result. However, all runs are typologically similar. Results for each single run can be found in Appendix D.1.1. Figure 6.7 shows the topology that all the results have in common. All the runs have two *legs* going towards the supports, a bar at the top and two *spokes* connecting the load to this structure. Some have an additional bar in the middle.

All the structures are sound; they resemble a common truss without the bottom elements. Those are not necessary since the supports can generate horizontal reaction forces. Two details draw attention. One is the bar in the middle that emerges in three of the eight generated structures. The pheromone on the elements that form this bar is never maximal. It seems like the bar has a very low stress, but that without it the structure would not be sound, possibly because the stress in the top bar would then become too big. The horizontal bars in the middle might have developed due to random fluctuations. Another interesting detail is the region where different members meet. The spokes seem to split into two in this area. An explanation for this could be that peak stresses occur near this connection of the spokes to the rest of the structure. Thickening at this point, as seen in the resulting structure of run 16, seems a good solution to cope with these peak stresses. For the other structures, to which a smaller load is applied, splitting the spoke into two is enough to provide a smooth flow of the forces from the spoke to the rest of the structure.

The best solutions found in the different runs are not identical. The runs have not fully converged by the 50th iteration when the optimisation process was stopped. It is noted that the parameter sensitivity runs were not yet performed at this stage in the algorithm development. It is likely that better results, or more similar results, are found once the parameter set is optimised.

One of the resulting structures will be compared in more detail to the Michell truss. The analytical Michell truss is optimised towards minimum weight with respect to the force it carries. In order to choose which of the generated structures is probably best comparable to the Michell truss, the load over volume ratio is calculated.

Table 6.1 shows that in theory, the resulting structure from run 16 should resemble the analytical Michell structure best. The resulting structure is shown in Figure 6.8. Compared to the Michell structure in Figure 6.2 and its BESO approximation in Figure 6.3, the result found with the proposed algorithm contains less *spokes* and a thickening at the location where multiple elements meet. Besides this, the topology is more or less identical. The differences can be easily explained.

The spokes transfer the loads from the location where the point load is attached to the structure to the *arch* spanning in between the supports. Because of the coarse meshing in the algorithm, the spokes have a minimum thickness that

Run	Load (kN)	Volume (m^3)	Ratio (kN/m^3)
6	400	0.196	2041
7	400	0.130	3077
8	400	0.122	3279
9	400	0.150	2667
10	400	0.136	2941
11	400	0.118	3390
12	400	0.146	2740
16	600	0.144	4167

Table 6.1: Load/weight ratio of coarse validation structures

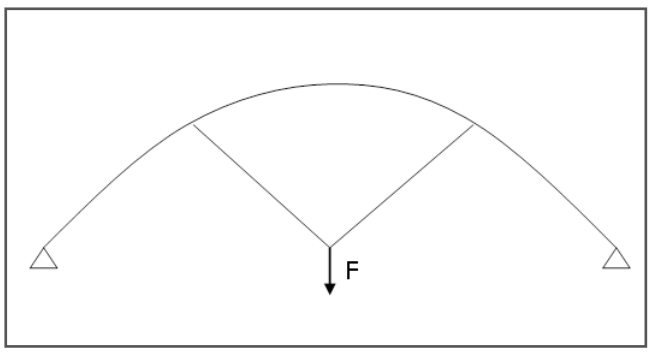


Figure 6.7: Topology analysis of coarse validation structures.

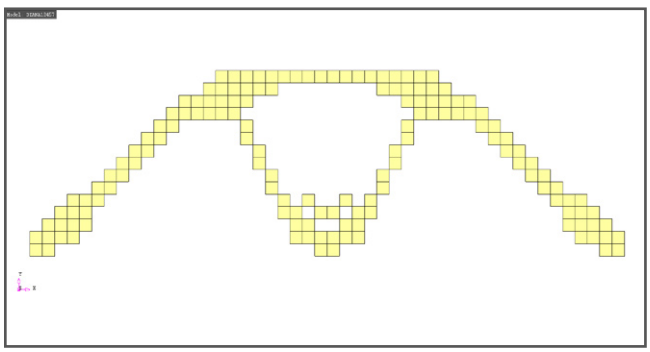


Figure 6.8: Resulting structure from run 16. Result mirrored along the line of symmetry.

is similar to the thickness of one element. Apparently, two spokes are enough to transfer the load to the arch. It can be expected that if the mesh is refined, more spokes will be needed and more spokes will develop.

The resulting structure from run 16 is manually checked according to the calculations as given in Appendix D.1.3 and Figure 6.9. Calculations show that for the given topology and the allowed stress of 50 MPa the cross sectional sizes as listed in Table 6.2 are needed.

The minimum cross sections are based on the element thickness of 100 mm as used in the mesh. According to the hand calculations, the cross section for member II suffices. Due to its orientation the member that results from the program has the exact 100 by 100 mm size. The other two members cannot have a clear cross section because their orientation is not in accordance with the orientation of the mesh. One can see that member I is supposed to be thicker than member III according to the hand calculation, which is also observed in the resulting structure in Figure 6.8. As a last remark it is noted that particularly the top member is stressed to a high level. It is therefore easy to understand that connections will be tricky due to peak stresses. In the resulting structure, one can observe that the connections are indeed thickened. The resulting structure from the algorithm is plausible.

6.1.2 Fine Michell truss

Validation runs have also been performed on a finer mesh for the same problem. Because these runs were performed in a later stage, a more advanced set of parameters was used and some extensions were used that were not yet used in the coarse validation runs. Only a few of the latest runs are presented. Detailed documentation is given in Appendix D.1.2. For all the runs presented in this paragraph, the actual runs are performed on half the structure. Symmetry is used to generate the complete structure from these results by mirroring.

Member	Minimum cross section
I	100*85 mm
II	100*100 mm
III	100*72 mm

Table 6.2: Minimum cross section for individual members

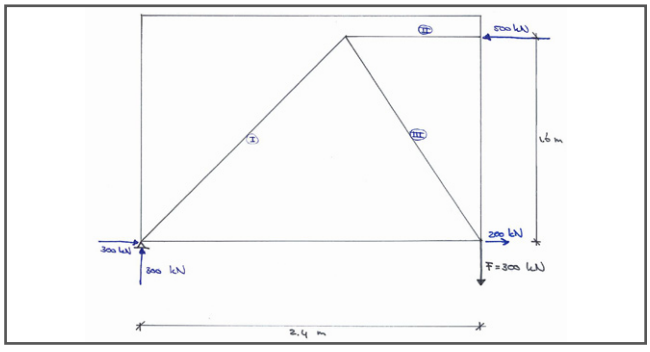


Figure 6.9: Model for manual checks.

Input

The mesh used in these runs is shown in Figure 6.10, while the boundary conditions are kept the same as in the coarse validation runs and are shown in Figure 6.4. The mesh now contains 48 by 32 elements; a total of 1536 elements and thus $2^{1536} = 2.41*10^{462}$ possible solutions. Elements have a thickness of 0.10 metres, and a length and a width of 0.05 metres.

It is chosen to take coarse run 16 as a starting point. This means the force applied to the middle of the structure is 600 kN. The material has the Young's modulus and Poisson radio of UHPC and a yield strength of 50 MPa. The maximal displacement is reduced to 0.010 metres, or span/480. Five runs are performed using this same input. They are presented in Appendix D.1.2.

Optimisation process

For these runs, the connectivity analysis was implemented along with the elitist strategy, solution copying and stress daemon. The holes analysis was not used. The performance function was limited to the following:

$$perf_j = \begin{cases} 1 \cdot 10^{-6} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-5} & if\ |\sigma|_{max} > f_y \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ \sigma_i \geq 0.05 f_y \\ perf_{i,k} + 0.9 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.05 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

These performance functions mean that the stresses and displacements have to stay within the allowable range, otherwise the performance of the generated structure is reduced to almost zero. A weak stress daemon applies only to elements with a very low stress.

The performance charts of the runs are given in Figure 6.11 and show that the runs that were performed had a very similar optimisation behaviour up to the 150th iteration. After that the optimisation runs start to diverge slightly. Apparently some of the runs have found structures that can not be optimised any further. None of the runs get badly stuck in a local optimum. However, the optimisation process was not yet finished by the 250th iteration when the process was stopped. These two phenomena can be explained by the weak

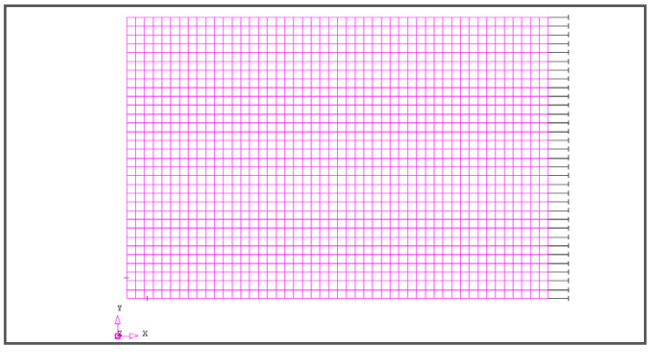


Figure 6.10: Mesh as used in the fine validation runs.

stress daemon, as will be explained in Paragraph 6.2.4.

Output

Just like the coarse approximation, the algorithm outputs structures that are supposed to be some kind of optimum in terms of light weight structures. First the solutions will be compared with each other and then the best solution will be compared to the Michell truss.

All of the solutions have an arch-like structure with a few connectors that transfer the point load to the arch. Especially the shape of the connectors changes between the solutions. Also, many small-scale discrepancies occur. The members of the structures are not very homogeneous. This problem with homogeneity can be explained. The algorithm does not have an incentive to make homogeneous elements. At the end of the day the volume of the structure is the only thing that counts. Because of randomness it is very likely that such inhomogeneous elements are constructed.

One of the resulting stuctures is compared to the Michell truss. The best developed solution in this series of runs is the structure shown in Figure 6.12. It is noted that the structure as shown in the figure is symmetrical because only half the structure was analysed and mirrored along its axis of symmetry. What was expected after the coarse validation runs proves to be true: the refinement in the mesh resolution allows for more *spokes* to emerge. The structure resembles the Michell truss in Figure 6.2 better than the coarse solution, and is almost identical to the BESO solution in Figure 6.3. This means that the optimal topology has been found.

At those places where members meet each other, extra material is placed in the structure. Just like in the coarse approximation, this is probably due to the flow of forces from one member to the other causing peak stresses in those regions.

In these runs the algorithm also shows its drawback. Because of its general approach and its selection of elements based on probabilities and randomness, inhomogeneous elements emerge that do not emerge with BESO. The fine approximation proved to be more sensitive for such random fluctuations, probably because the search space is bigger.

The members that are shown in the resulting structure in Figure 6.12 are about the same size as the members found in the coarse validation runs. The only difference is formed by the spokes. Now two thin spokes emerge rather than one thicker one. Member sizes have been checked on the results of the coarse runs and because the member sizes in the fine runs are similar for the same boundary conditions, they will not be checked again and will be assumed to suffice.

6.1.3 Validating the calculations

Up until this point, it has been assumed that the finite element package calculates in a correct way. Here, a bit more insight will be created in these calculations. As a verification example, the problem shown in Figure 6.13 has been optimised for. The optimisation objectives were as follows.

- Optimisation objective: minimum weight.
- Allowable stress: 80 MPa in tension as well as compression.
- Allowable deflection: 0.50 metres.
- Span: 105 metres.
- Element thickness: 0.10 metres.
- Load: 25 kN/m[1].

Only half the structure was analysed; a space of 52.5 by 10

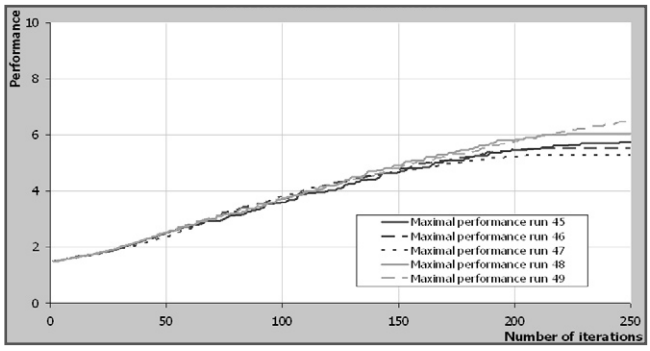


Figure 6.11: Performance charts of fine validation runs.

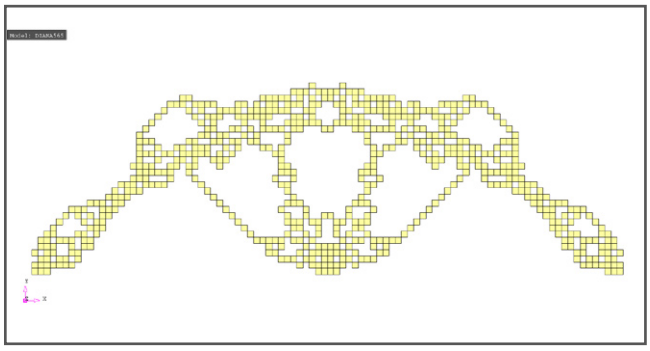


Figure 6.12: Best solution found for fine validation runs. Result mirrored along the line of symmetry.

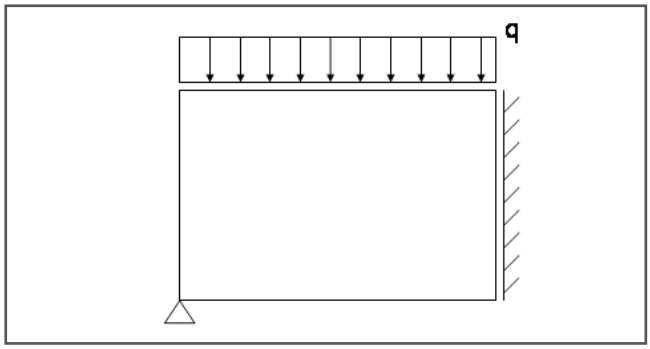


Figure 6.13: Boundary conditions for calculations verifications.

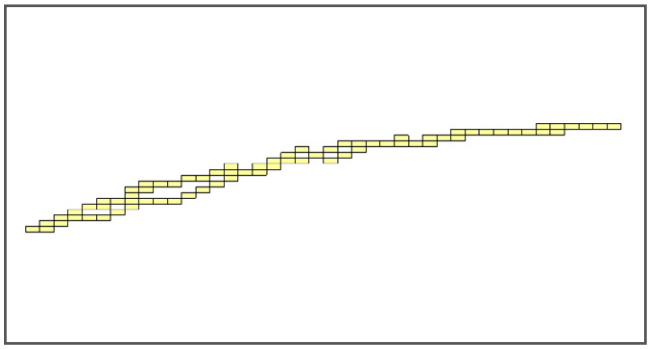


Figure 6.14: Best solution found for calculations verifications run.

metres. A 42 by 20 element mesh was used. The run resulted in the structure shown in Figure 6.14. The structure is an arch, which is in indeed the optimal shape for such a structure. All the checks in this paragraph have been performed on half this structure that directly followed from the optimisation process.

Analysis of the voids

In the optimisation process ants can choose to put material in an element or leave it empty. In the finite element calculations, it is very difficult to work with a truly empty element. In order to still be able to leave elements 'empty', the proposed algorithm uses a very weak material at the place of these voids. The material properties of this material are:

- E_{void} = 1.0 Pa
- ν_{void} = 0.0 (-)

The only function of this material is to transfer the loads from the place where they are applied to the mesh to the structural element. In Figure 6.15 the deformation of the material that represents the voids is depicted. In this case, the load is applied to the top of the mesh. The scale factor applied to the deformation plot is very small: $0.5*10^{-5}$. This means the material deforms considerably. It does not have any stiffness to cope with the loads itself. The deformed shape is as one would expect for a sponge. It will transfer the loads to the elements that support the sponge: the concrete shell. The material underneath the shell is not loaded and does not deform. At first glance, the material seems to behave as it is supposed to.

Analysis of the material

The stresses in the elements are fairly low, and can be checked manually. The following data applies:

- Structural depth: 10 metres.
- Span: 105 metres.
- Load: 9.6 kN/m[1]
- Element surface: $0.1*0.5 = 0.05$ m^2.
- Bending moment in the middle of the span: $1/8*q*l^2 = 13$ MNm
- Force in shell at the middle of the span: $M/h = 1.3$ MN
- Stress in shell at the middle of the span: $F/A = 26.5$ MPa

The difference between the load in this calculation and the load implemented in the algorithm occurs because of the way the algorithm works. The algorithm works with a standard load (mostly equal to 1 kN/m^2) which is then multiplied with any load that is considered. In this example, a load case of 2.56 kN/m^2 is considered for an arch that carries the load of a strip with a width of 9.6 metres. Within the algorithm, the average load per element is multiplied with the considered load. This multiplication can be performed because the stresses are within the linear elastic range.

In this case, the stress in the shell at the middle of the span should be 26.5 MPa for the 1 kN/m^2 over the width of the strip. Multiplied with the load case (2.56 kN/m^2) the stress in the shell for this load becomes 67.8 MPa. This is lower than the stress limit that is set to 80 MPa. The shell has a curvature: in theory, the force in the shell should grow slightly closer to the supports. This leads to either a larger stress or a thicker shell.

In Figure 6.16 the stresses according to the FEM analysis are plotted. The minimum stresses, or the largest compressive stresses, are mostly around 20 to 30 MPa, just as calculated. However, extreme values are obtained in corners of the elements. These extremes are compensated for on purpose within the algorithm, as will be explained.

The shape of the element is not equal to the shape of the real structure once built; due to the meshing it consists of all squares. If one of those squares is slightly eccentrical, a bending moment is introduced into the element. In DIANA, stresses are calculated in integration points within the elements that are shown in Figure 5.5. This will cause different stresses in the different integration points. However, when the structure will be built, the discrepancies that occur due to the meshing will not occur. Therefore the bending moments in the elements will be reduced. In this stage of designing, it is therefore more accurate to take only the average of the stresses in the four integration points into account. This stress approximates the stress in the structure best.

There is another phenomenon that causes the stresses in the post-processor to look even more extreme. The post-processor that comes with DIANA interpolates the stresses towards the edges of the elements. The extreme stresses shown in the post-processor are therefore not equal to the stresses that are calculated in the integration points [7]. In short, the bending moments introduced due to the meshing will not occur in the real structure and the difference in stresses they cause in the different integration points are exaggerated in the post-processor that comes with DIANA.

To correctly interpret Figure 6.16 one should compare the average stress within an element with the value that is manually calculated. This is the reason why the minimum stresses that are shown in the plot of around 20 to 30 MPa on average

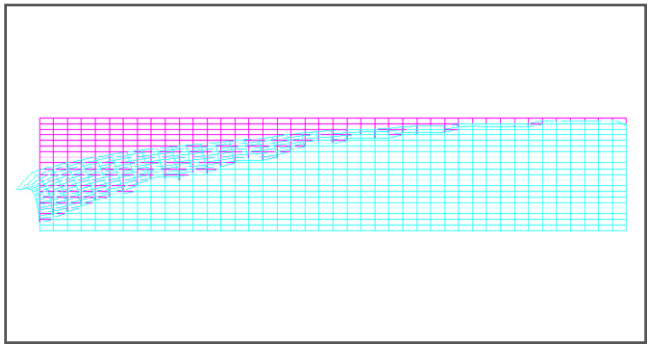


Figure 6.15: Deformation of the material representing the voids. Scale factor $0.5*10^{-5}$.

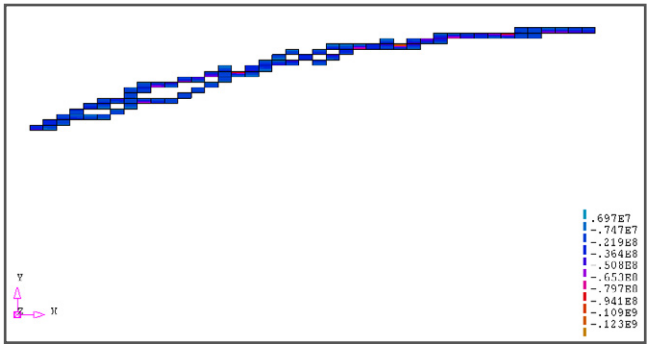


Figure 6.16: Stresses in the best solution in Pa (N/m^2).

are satisfying.

The deformations of the structure are shown qualitatively in Figure 6.17 and quantitatively in Figure 6.18. Because of the same load factorising as explained earlier, the results should be multiplied with the load case (2.56 kN/m²). All displacements are well within the set range. The qualitative displacement graph shows that the displacement is big at a point where two elements only touch each other's corner. The elements rotate around this corner and a big displacement is found. This is another discrepancy due to the meshing but is not worrying. If a connection with a bending moment is needed, the algorithm is capable of assigning material to a connecting element such that this displacement will not occur.

The validation of the elements to which material is assigned also validates the assumption done for the elements that were left 'empty' by the ants; the forces applied to the top of the mesh are properly transferred to the actual structure.

6.2 Parameter sensitivity

Extensive tests are performed in order to find out how the optimisation process depends on the different parameters that have to be set at the start of the run. Several sets of parameters are compared to one initial model. Each set of parameters has only one different parameter in comparison with the initial set. The initial parameter set or *base parameter set* is a set of parameters that has proven to be well-functioning over the numerous test runs that are performed whilst developing the algorithm.

In this paragraph only the convergence of the runs is discussed. More extensive run data, including the complete set of parameters, the best solutions found during the process and the resulting best structure are given in Appendix D.2.

Analysed problem

Because the algorithm will be applied to the roof of the Omnisport building, this problem has been used for the sensitivity analysis. The mesh and the boundary conditions that are used are given in Appendix D.2. The symmetric nature of the problem is used by analysing only half the structure. All the plots are mirrored along the line of symmetry. The full algorithm with all proposed extensions is used. The performance function to which the optimisation process is aimed is the following.

$$perf_j = \begin{cases} 1 \cdot 10^{-6} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-5} & if\ \sigma_{max} > \sigma_{allowable;max} \\ 1 \cdot 10^{-5} & if\ \sigma_{min} < \sigma_{allowable;min} \\ 2 \cdot 10^{-5} & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1}{2V + 0.3S + 0.5PV_p + 20H} & else \end{cases}$$

Where:

$perf_j$ = Performance of structure j.
V = Volume of the structure (m³).
S = Surface area of the structure (m²).
P = Pre-stress level (-).
V_p = Volume of pre-stressed elements (m³).
H = Level of matching the holes criterion.

The constant for taking into account the holes variable is set to 20.

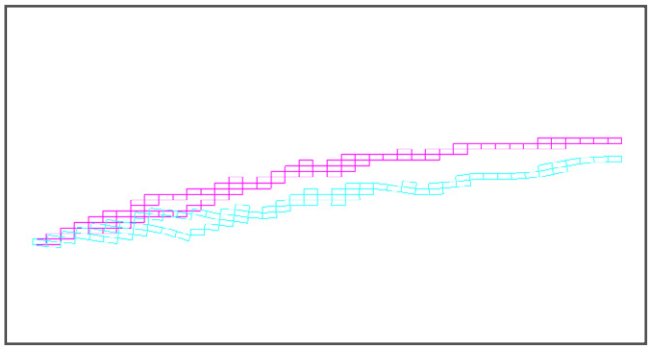


Figure 6.17: Qualitative displacements in the best solution. Scale factor 20.4.

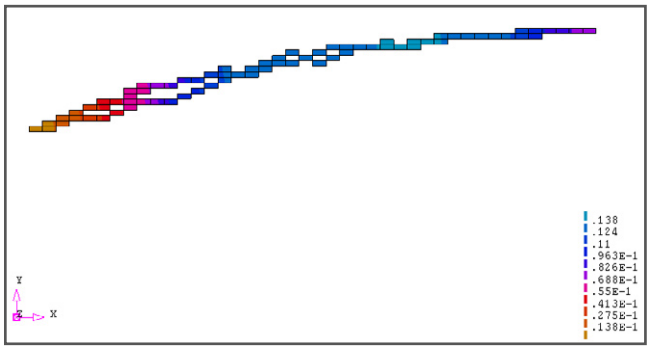


Figure 6.18: Quantitative displacement in the best solution in m.

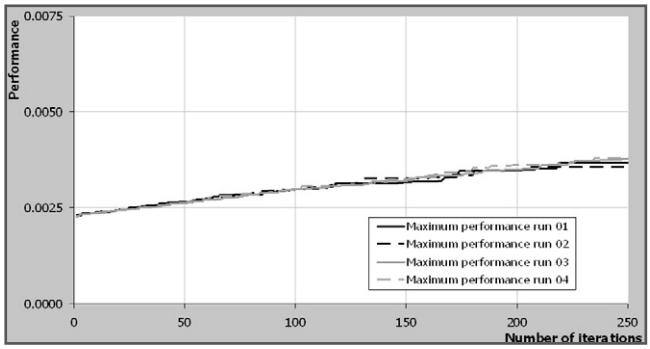


Figure 6.19: Performance chart for the four runs with pheromone decay factor 0.05.

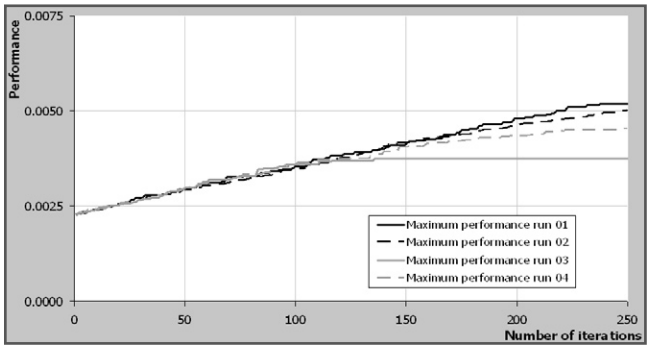


Figure 6.20: Performance chart for the four runs with pheromone decay factor 0.10.

Because allowing tensile stresses with pre-stressing means the compressive capacity goes down as explained in Paragraph 5.2.4, the stresses that occur in the same direction need to stay within the range specified. If these boundary conditions are fulfilled, as well as the boundary condition with respect to displacement, the structure is assigned a performance that is the inverse of its estimated costs. This performance function will be analysed later in Paragraph 6.3.

The maximum and minimum allowable stresses for the runs that form part of the parameter sensitivity analysis are set to 80 MPa and -80 MPa, respectively. These values are based on the linear elastic range determined in Paragraph 4.4.2. In that paragraph it is shown that the compressive stress in the ULS can be as high as 102 MPa. Due to pre-stressing, this value could in theory also be obtained for tensile stresses. Pre-stressing will be needed for tensile stresses higher than 5 MPa. The allowable stresses are reduced to 80 MPa because of buckling for compressive stresses and pre-stress losses for tensile stresses.

Random fluctuations

The proposed algorithm generates solutions using a probability distribution and random numbers. An effect of the randomness is that every run yields a slightly different result. Also, some runs will show premature stagnation (i.e. get stuck in local optima) while other runs with the same set of parameters will not. To filter out the influence of this randomness, each run is performed four times. The best run out of this set of four is used as an example of what the possibilities of the used set of parameters actually are.

6.2.1 Pheromone decay factor

The pheromone decay factor ρ determines to what extent newly developed solutions influence the pheromone distribution. The pheromone distribution is the collective memory of all the previous runs. The following function defines the pheromone update process.

$$RV_i(k+1) = (1-\rho)RV_i(k) + \rho\frac{perf_{i,k}}{perf_{\max,k}}$$

In the first term, *(1-ρ)* determines how much pheromone from iteration *k*, defined as $RV_i(k)$, is copied directly into the pheromone distribution that is used for the next iteration. In the second term ρ defines how large the contribution is of the performance of an element in iteration *k* to the pheromone distribution for the next iteration *(k+1)*.

A large pheromone decay factor makes the algorithm greedy; newly developed solutions highly influence the chance that parts of that solutions will be chosen again in later iterations. One can expect that the algorithm will then quickly converge to a solution. Taking a small pheromone decay factor, the newly developed solutions will have a relatively small influence. The pheromone will *evaporate* less quickly, and old solutions will influence new solutions over a bigger amount of iterations. It can be expected that convergence is now slower, but that the search space is searched more carefully.

Pheromone decay factor 0.05

When a very small pheromone decay factor is chosen and implemented in the algorithm, the optimisation process is slow. Figure 6,19 shows the development of the performance of the best structure over the iterations. As can be seen in the

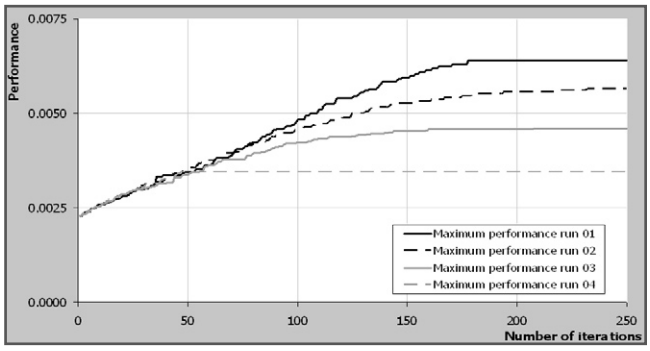


Figure 6.21: Performance chart for the four runs with pheromone decay factor 0.20.

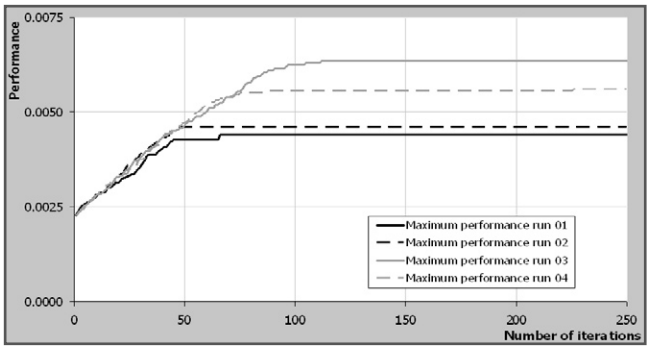


Figure 6.22: Performance chart for the four runs with pheromone decay factor 0.40.

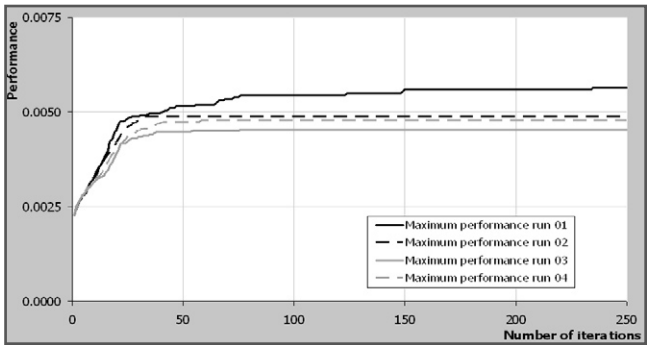


Figure 6.23: Performance chart for the four runs with pheromone decay factor 0.80.

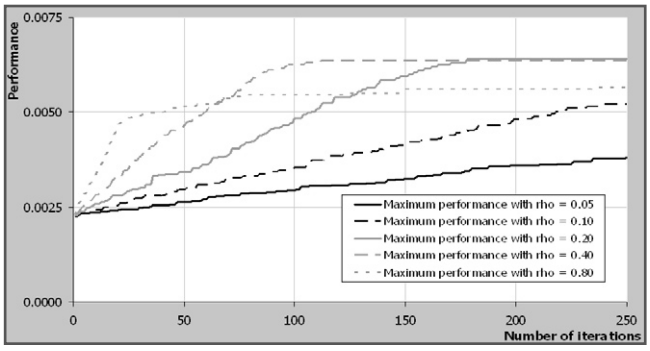


Figure 6.24: Performance chart for the best runs with different pheromone decay factors.

figure, the optimisation process is still ongoing at the end of the run. None of the runs have reached stagnation.

Pheromone decay factor 0.10

Increasing the pheromone decay factor increases the speed of optimisation. However, also the risk of getting stuck in a local optima increases slightly. One of the runs, as can be seen in the performance chart in Figure 6.20, did get stuck in a local optimum. The other runs are still finding better solutions after 250 iterations. Clearly, 250 iterations is not enough for the process to complete with such a small pheromone decay factor.

Pheromone decay factor 0.20

The characteristics of a growing pheromone decay factor become more clear in the runs with a pheromone decay factor of 0.20, depicted in Figure 6.21. One of the run stagnates in a local optimum, while the optima found by the other runs are significantly different. The more greedy behaviour makes the sensitivity to random fluctuations bigger.

Pheromone decay factor 0.40

Figure 6.22 shows that this pattern is extended to the runs with a pheromone decay factor of 0.40. Quicker convergence occurs in the beginning of the runs and some runs get stuck in non-optimal solutions. One of the four runs generates a very well performing structure.

Pheromone decay factor 0.80

The case of this extreme pheromone decay factor is researched and results are shown in Figure 6.23. One can observe the very greedy behaviour and the stagnation in non-optimal solutions in all the runs.

Conclusion on the pheromone decay factor

In Figure 6.24 the combined performance charts of the best runs from each of the parameter sets is given. It can be concluded that using a large factor leads to rapid optimisation over iterations, whereas a small factor leads to slow optimisation over iterations. However, the rapidly optimised runs have a high probability of giving non-optimal results. Runs with a small pheromone decay factor optimise slower, but eventually find better solutions. The runs in which pheromone decay factors of 0,10 and 0,05 are used are not yet converged to an optimum. It can be expected that when the number of iterations is increased, these runs will approach even better optima than found in the other runs. There is a direct link between the pheromone decay factor and the number of iterations needed. To limit the number of iterations and make the runs faster, the pheromone decay factor should be increased up to the point when it gets too sensitive for getting stuck in non-optimal solutions. In this case, a pheromone decay factor of 0.40 proves a good choice, but it is advised to be a bit more conservative and use 0.20 because its results are more reliable.

6.2.2 Number of ants

The number of ants equals the number of solutions generated in each iteration. It therefore determines how well the search space is searched in one iteration. Using more ants increases the chance of finding a proper optimum but slows down the process; the run time needed by the algorithm is linearly dependent on the number of ants used if the number

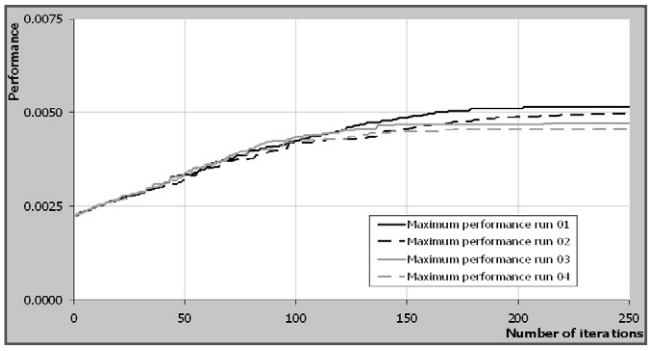


Figure 6.25: Performance chart for the four runs with 25 ants.

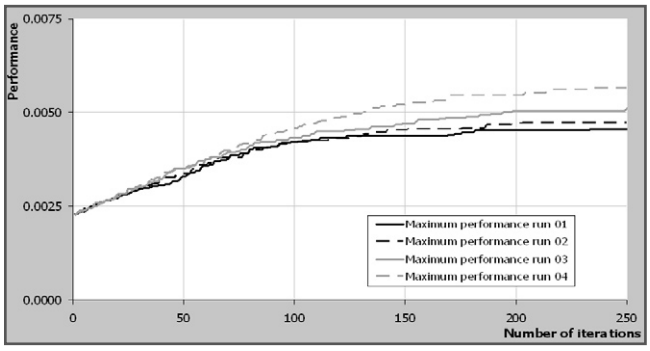


Figure 6.26: Performance chart for the four runs with 50 ants.

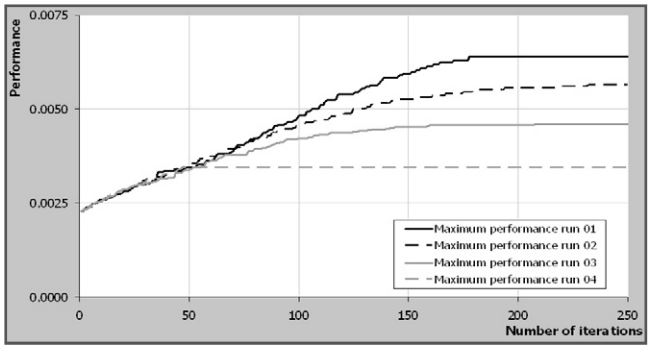


Figure 6.27: Performance chart for the four runs with 100 ants.

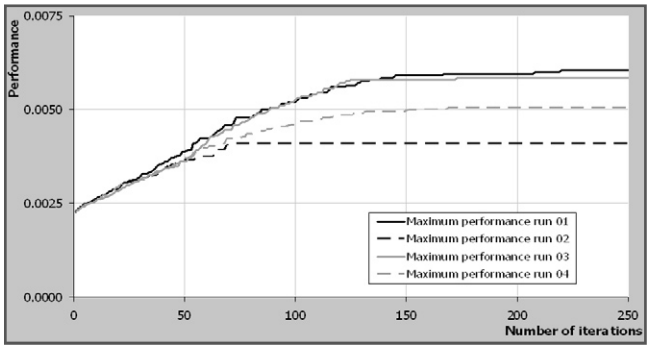


Figure 6.28: Performance chart for the four runs with 250 ants.

of iterations is kept the same.

25 Ants

One could expect that using only 25 ants, the results are highly subjective to the variances in the probabilistic decisions the few ants make. In Figure 6.25 it can be observed that the performance of the generated structures is in fact quite similar. The structures themselves, shown in Appendix D.2.2, are fairly similar. It seems that the metaheuristic approach in the algorithm is robust, even with a small number of trial solutions.

50 Ants

When the number of trial solutions is increased (Figure 6.26), the runs optimise slightly quicker. This is, they need less iterations to come to an optimum. However, the run time is longer because the run time per iteration is linearly dependent on the number of ants used. The routine finds slightly better solutions at the end of the run. The difference in the performance of the best solutions found in the different runs is bigger than when using 25 ants.

100 Ants

This pattern is even more clear when considering the runs with 100 ants (Figure 6.27). Some runs find superior solutions, just because more possible options are tried during the process and solutions that have a smaller chance of getting chosen sometimes emerge. One run gets stuck in a local optimum, possibly due to random fluctuations.

250 Ants

Figure 6.28 shows that the run with 250 ants optimises quicker over iterations in the first few iterations in comparison to runs with less ants. It finds an optimum that is fairly similar to the one found in the 100 ant run. One of the runs stagnates prematurely.

500 Ants

Increasing the number of ants further to 500 does not increase the performance accordingly (Figure 6.29). None of the runs gets stuck in a local optimum, which can be because of the very exploratory behaviour, or just because of the luck factor in the process.

Conclusion on the number of ants

The combined convergence charts for the runs with different numbers of ants is given in Figure 6.30. Increasing the number of ants leads to a slightly quicker optimisation in terms of iterations needed to find an optimum. Also, the search space is explored better. This results in runs with more ants finding better optima. This phenomenon is only visible when letting the ant population grow to 100 ants. From that number on, using more ants does not lead to better optima anymore.

6.2.3 Size of elite

Only the best solutions are used for the determination of the pheromone distributions in the next iteration. These solutions are part of the *elite*. The elitist solutions contribute to the pheromone distribution according to their performance. However, the difference in the performance of the elitist solutions is often so small that one can say that their contribution is similar.

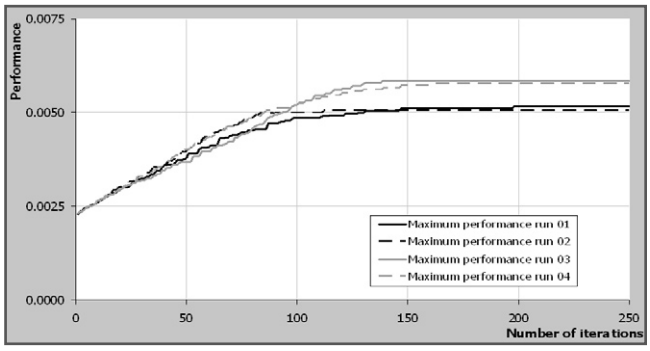


Figure 6.29: Performance chart for the four runs with 500 ants.

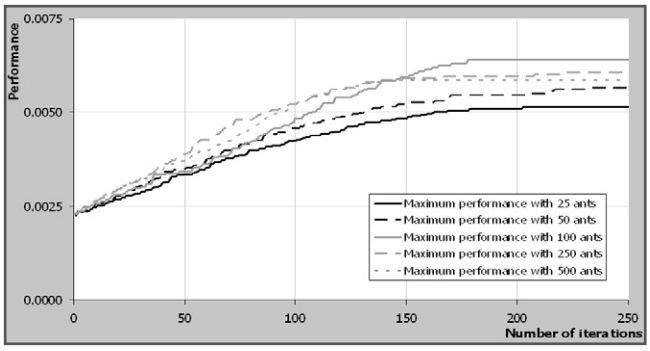


Figure 6.30: Performance chart for the best runs with different numbers of ants.

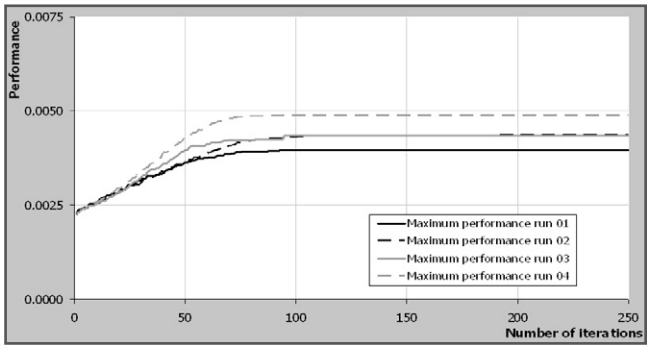


Figure 6.31: Performance chart for the four runs with 1 elitist ant.

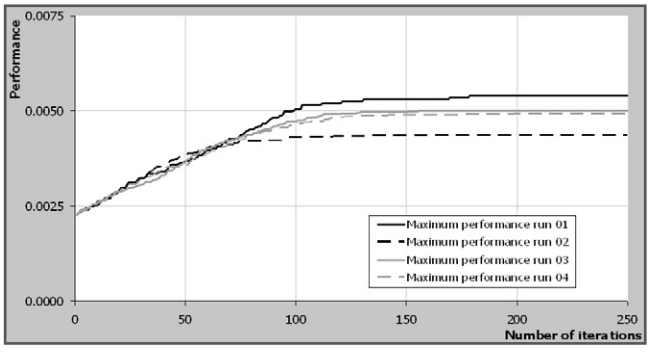


Figure 6.32: Performance chart for the four runs with 2 elitist ants.

It can be expected that a small elite makes the algorithm greedy: only the elements that were part of the best solution are assigned pheromone, and the ants will quickly converge to this one solution. Using a large elite means that many solutions are assigned pheromone at the end of an iteration. The process will keep the options open for a longer period of time.

1 Elitist ant

Figure 6.31 shows the performance charts of the runs with one elitist ant. The runs optimise quickly but the exploratory behaviour stops after about 75 iterations and the algorithm is not capable anymore of finding new, better solutions. This could have been expected as explained above.

2 Elitist ants

When two elitist ants are allowed to lay pheromone, optimisation is slowed down but exploratory behaviour increases and the ants find slightly better but not entirely optimal solutions, as shown in Figure 6.32.

5 Elitist ants

Increasing the number of elitist ants further (Figure 6.33) leads to a better optimising behaviour for most runs. The search space seems to be better explored and the routine is able of selecting a fairly good optimum.

10 Elitist ants

Doubling the size of the elite and going to 10 elitist ants yields the performance graph in Figure 6.34. Some runs now get stuck in a local optimum. This can be explained by the very exploratory behaviour of the ants. When assigning pheromone to many solutions, there is a risk that many elements get assigned a little bit of pheromone. To find a structure that works a completely connected structure is needed. This means that in a solution enough elements have to be selected to form a complete structure. When many elements have a chance of getting assigned material of say 70%, there is a risk that many structures are generated that miss vital links. This can explain the stagnating behaviour of some runs. In runs where this phenomenon does not occur, the exploratory behaviour makes the routine find better solutions than in the runs performed with five elitist ants.

20 Elitist ants

When doubling the size of the elite again to 20 elitist ants the performance graph in Figure 6.35 evolves. The large elite makes the optimisation process slow but steady. The runs are fairly similar, except for one that gets stuck in a local optimum, which can be explained by the reasons listed for the runs with 10 elitist ants.

The resulting structures after 250 iterations are given in Appendix D.2.3. It is noted that the three non-stagnating runs yield very similar optimal structures. This confirms the proper optimising behaviour in this run with a large elite.

Conclusion on the size of the elite

The size of the elite has proven to be an important parameter in the algorithm. When the elite is small, the optimisation process is quick but the search space is not well searched. Early found optima are quickly approached and if the optimum happens to be local, the algorithm does not have the flexibility to climb out of this optimum.

A large elite keeps the run from choosing one area in the

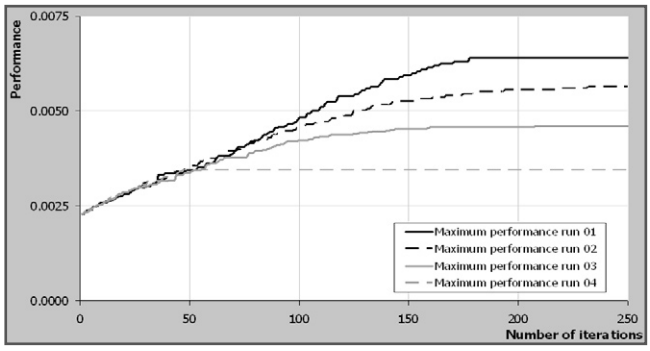


Figure 6.33: Performance chart for the four runs with 5 elitist ants.

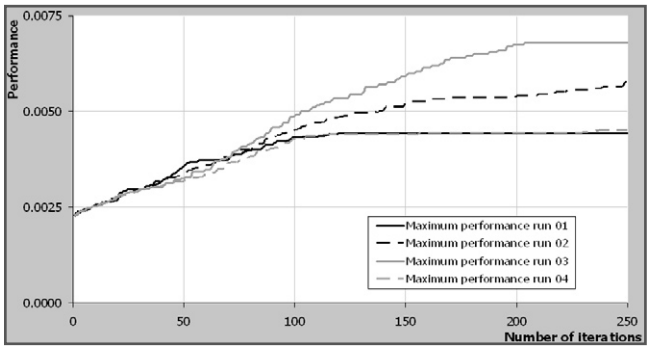


Figure 6.34: Performance chart for the four runs with 10 elitist ants.

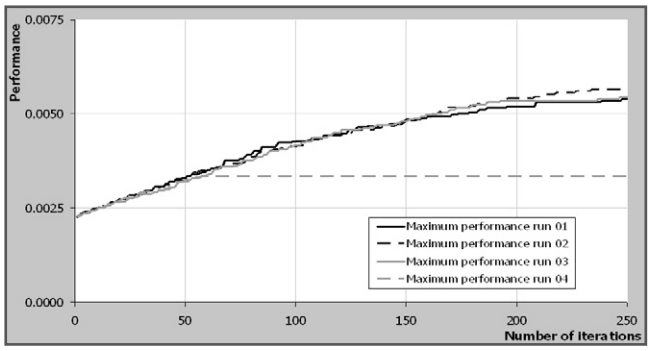


Figure 6.35: Performance chart for the four runs with 20 elitist ants.

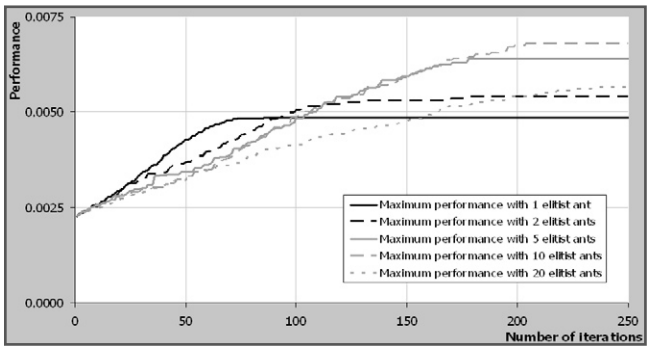


Figure 6.36: Performance chart for the best runs with different numbers of elitist ants.

search space to explore further. Instead, many elements will be assigned a reasonable amount of pheromone. Now another risk emerges. Structures might be generated that are not complete, because the reference value of an element will not approach the number one. This phenomenon is possibly connected with the stress daemon that will be researched next.

The performance graph of the best runs for each set of parameters is given in Figure 6.36. In this graph the influence of this parameter on the speed of optimisation becomes clear. Also one can see that the optimum found with one or two elitist ants is not quite optimal, while runs with 20 elitist ants did not stagnate within 250 runs. The runs with 5 or 10 elitist ants give the best solutions. For this reason it is recommended to use a number of elitist ants in between 5 and 10.

6.2.4 Stress daemon

A stress daemon action guides the search performed by the routine. Elements that are part of a solution but only have a small stress are not assigned the full amount of pheromone, but a fraction. By doing so the effect of a *rejection list* is obtained: the routine is less likely to assign material to these elements in a next generation, but the chance still exists.

In the routine two stress thresholds are used: one at 1% and one at 5% of the maximally allowable stress. It is chosen not to vary these thresholds but only the performance penalty connected with them.

Performance adjustment of 0.50

In these runs, the performance of elements that have a stress lower than 5% of the yield stress is multiplied with a factor 0.50. In the performance chart shown in Figure 6.37 the effect of this action is clarified. The performance goes up quickly because many elements are rejected early in the process. The drawback of the stress daemon action is that elements that are needed to generate a completely connected structure have a low probability of getting selected, which leads to many incomplete structures. When only incomplete structures are generated, the best-so-far structure is kept as the optimal structure and the run stagnates prematurely.

Performance adjustment of 0.50 and 0.70

In the runs described here the performance of elements with a stress lower than 1% of the yield strength is limited to *0.50*performance*, while the pheromone of elements with a stress lower than 5% of the yield strength is limited to *0.70*performance*.

This leads to a similar performance chart as the runs with a pheromone limit 0.50 for both thresholds; see Figure 6.38.

Performance adjustment of 0.50 and 0.90

The performance in these tests is multiplied with 0.50 and 0.90 for elements with a stress lower than 1% and 5% of the yield stress, respectively and is plotted in Figure 6.39. Still some runs stagnate but also some very good runs are performed leading to satisfying structures.

Performance adjustment of 0.70

Setting both performance adjustment factors to 0.70, the performance graphs are not very often interrupted (Figure 6.40). Fairly well performing structures are obtained.

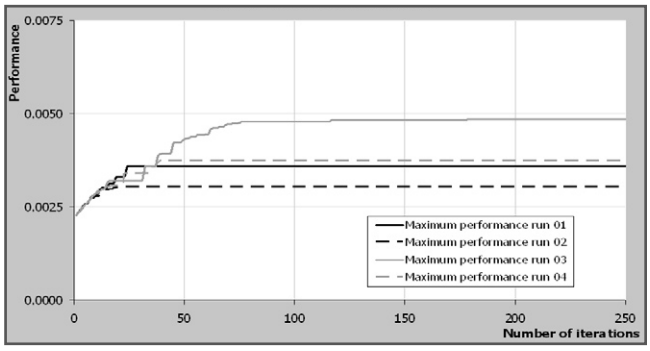


Figure 6.37: Performance chart for one stress threshold at 0.50.

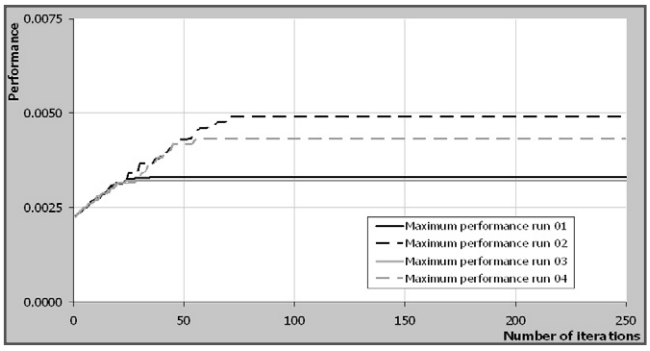


Figure 6.38: Performance chart for two stress thresholds at 0.50 and 0.70.

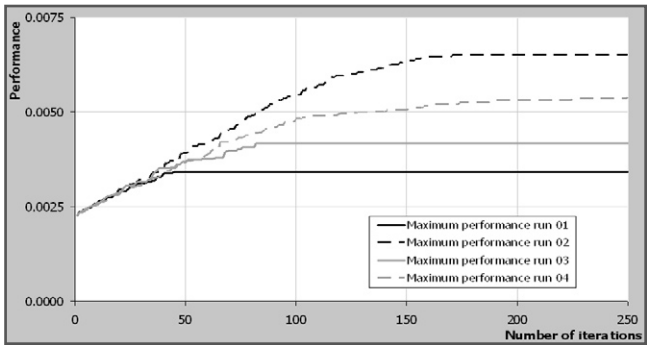


Figure 6.39: Performance chart for two stress thresholds at 0.50 and 0.90.

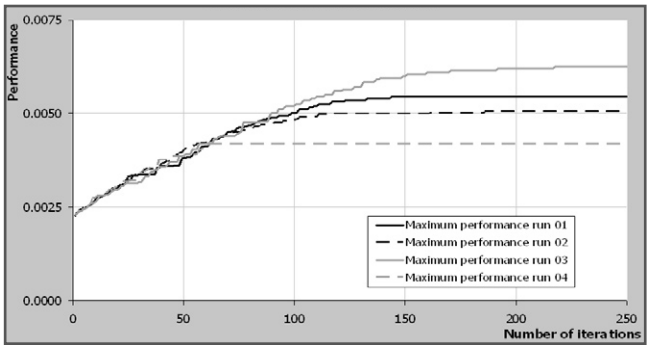


Figure 6.40: Performance chart for one stress threshold at 0.70.

Performance adjustment of 0.70 and 0.90

For these runs more or less the same conclusions as the previous test parameters can be drawn. The convergence chart in Figure 6.41 shows one stagnating run and quite a big difference in the end-of-run cost of the other runs.

Performance adjustment of 0.90

As a last test runs have been performed where the influence of the stress daemon was negligible. Elements with a stress lower than 5% of the yield stress were given a performance reduction to 0.90 times the original performance. In the resulting performance charts, given in Figure 6.42, the convergence of the runs proves to be very similar.

Conclusion on the stress daemon

Using the stress daemon can lead to an increase of prematurely stagnating runs. One could also say that the larger the influence of the daemon action, the larger the chance that a run is poor due to the randomness in the routine. The stress daemon makes the algorithm less *robust*. This is confirmed by the last run with only a very small influence of the stress level on the performance. The results of these runs are very similar.

Why then would one introduce the stress daemon in the first place when it decreases the robustness of the algorithm? Figure 6.43 shows that the convergence of the runs with a stress daemon is quicker, especially in the beginning of the process. This can lead to superior results, as the run with a stress daemon of 0.50 – 0.90 shows. The stress daemon is a very good guide because it guides the search for the best solution in a structural way and works as a kind of rejection list. The influence on the robustness, however, is very large and therefore it can be concluded that the stress daemon is a good feature of the proposed algorithm but should be handled with care because it comes with the risk of generating incomplete structures. The test runs show that the stress daemon is possibly the main reason for runs to stagnate.

6.2.5 Number of iterations

The speed of optimisation and therefore the number of iterations needed, depends on the parameters that are chosen. An *eager* set of parameters can generate well-performing structures quickly but comes with the chance of not finding the global optimum. A less eager run needs more iterations to converge. It is noted that the process is fairly time-consuming. Most of the test runs took about 105 minutes. A lot of time can be saved by choosing such a set of parameters that not too many iterations need to be made. This choice, however, is up to the user of the routine.

6.2.6 Pheromone limits

The proposed algorithm provides the option of limiting the reference value of the elements to a certain number. Limiting it to a number in between 0.05 and 0.99 for example, means that there is a minimum chance of 5% and a maximum chance of 99% that the element gets assigned material. This improves the chance of finding new better optima later on in the process.

Pheromone limits 0.05 and 0.99

Test runs have been performed in which the reference value was limited to 0.05 and 0.99. A run of 250 iterations has been

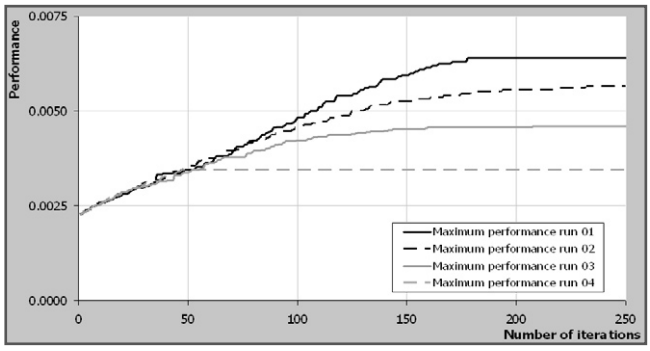


Figure 6.41: Performance chart for two stress thresholds at 0.70 and 0.90.

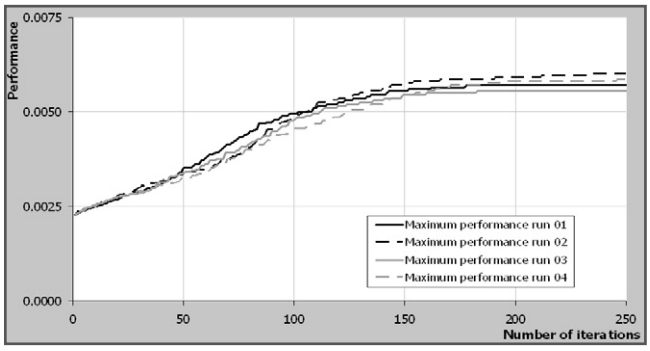


Figure 6.42: Performance chart for one stress threshold at 0.90.

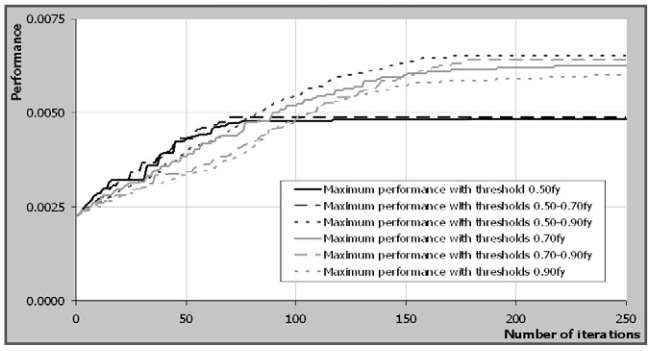


Figure 6.43: Performance chart for the best runs with different stress thresholds.

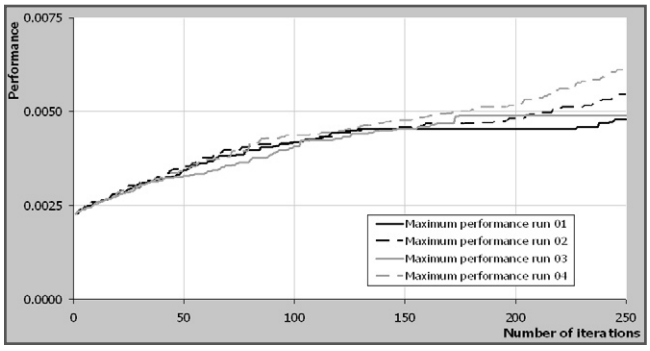


Figure 6.44: Performance chart for runs with reference value limits.

performed. After 200 iterations, the reference values have been reset to 0 and 1. The performance chart, given as Figure 6.44, shows optimisation in two steps. In the first 200 iterations, the routine creates structures that are near-optimal but also use 5% of the badly performing elements. After 200 iterations, also these badly performing elements that are selected due to the 5% chance of being chosen start to be rejected as well.

Because the second part of the optimisation process starts after 200 iterations the algorithm still finds better solutions at the end of the run. The process therefore is not completely stagnated. Interestingly, one can also see that one of the five runs did not increase the performance of the best structure after 200 iterations.

No pheromone limits

Figure 6.45 shows the performance chart of four runs without limits in the reference value. Some runs stagnate prematurely but other runs that show satisfying optimisation behaviour obtain a better end-of-run performance than the runs with pheromone limits.

Conclusions on pheromone limits

The implementation of pheromone limits has the advantage that less runs get stuck in a local optimum. However, results that are found are not necessarily better and because the optimisation process *restarts* after a certain number of iterations, it takes more iterations to end the run. It seems that adding the limits in the pheromone values does not have a major influence and merely complicates the optimising behaviour making it more difficult for the user to get a feel of what happens. However, the pheromone limits can be used if runs on a problem stagnate very often. In the case of using the pheromone limits, it is recommended to use very small adjustments to the pheromone limits like 0.005 and 0.995.

6.2.7 Number of elements

It is interesting to know what the influence is of the size of the problem on the time needed for the optimisation process. The number of elements has a two-fold influence on the optimisation run time.

- The calculation time of a finite element calculation grows if the mesh contains more elements.
- The search space for the algorithm grows with the number of elements.

Several meshes with different numbers of elements have been analysed during the development of the algorithm. In Figure 6.46 the time needed for one *ant run*, which consists of the generation and analysis of one structure is plotted against the number of elements that are used in the mesh. The average run times per mesh are an everage of several runs. According to the results a linear dependency is assumed. This dependency can not be concluded with a high level of certainty, because only a few different meshes have been analyses thoroughly in the algorithm.

In Figure 6.47 the number of ant runs that the algorithm needs before it finds an optimum are set out against the number of elements in the mesh. It is noted that this is done in a fairly rough manner. The number of ant runs is not only dependent on the number of elements, but also on the parameter set that is used, as is explained earlier in this paragraph. A more or less linear dependency can be assumed when analysing the graph. This dependency cannot be vali-

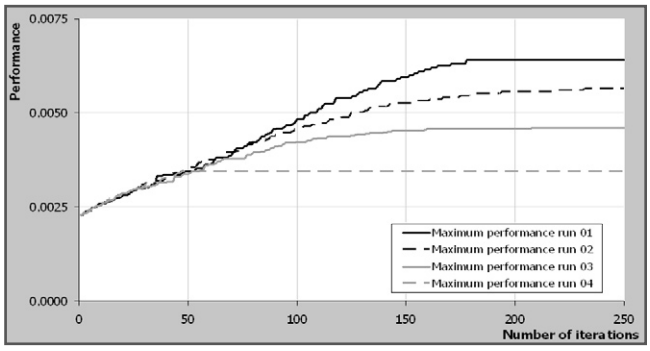


Figure 6.45: Performance chart for runs without reference value limits.

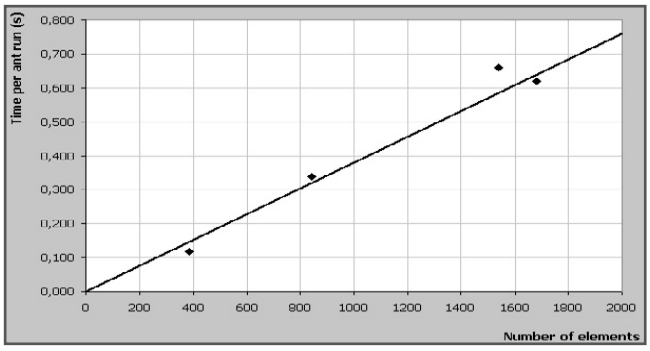


Figure 6.46: Time needed for one ant run depends on the number of elements in the mesh.

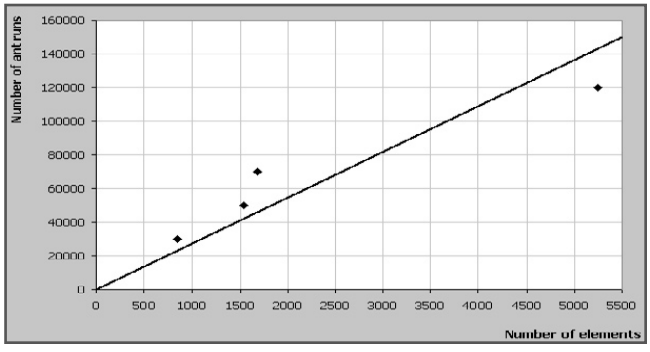


Figure 6.47: The number of ant runs needed until an optimum is found.

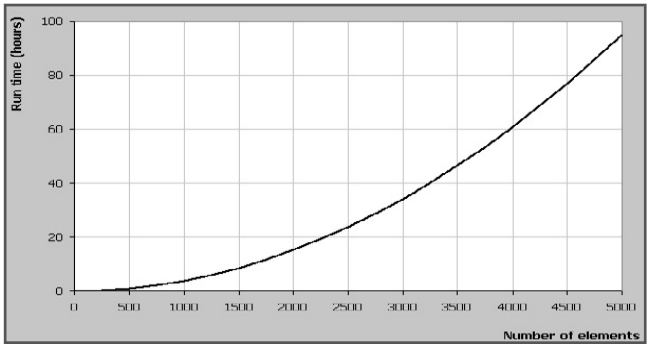


Figure 6.48: Estimated total run time as a function of the number of elements in the mesh.

dated based on the relatively small amount of data available at this point.

The two linear dependencies that are assumed make the dependency of the run time needed for the algorithm to converge to a solution quadratic. In Figure 6.48 the estimated total run time in hours is plotted as a function of the number of elements in the mesh. This assumed dependency is:

$T_{process} \approx 0.014 * n_{elements}^{\ 2}$

Where $T_{process}$ is the time in seconds.

It is explicitly noted that this is an assumed dependency that cannot be verified at this point. It is merely in indication of the run time one can expect when setting up an optimisation process using the proposed algorithm.

6.2.8 Recommended set of parameters

For each parameter that has been analysed in this paragraph it was concluded that it gives the user the choice between a quickly converging process or a larger chance of getting premature stagnation.

The factor determining the rate of decay of the pheromone is recommended to be set around 0.20. The factor 0.40 also led to good results, but lower values make the process very slow.

The number of ants is recommended to be set around 100. Increasing this number linearly increases the run time.

The size of the elite can best be set around 5 or 10. Dorigo et al. [27] conclude an optimal size of the elite of eight elitist ants. This study confirms this number.

Limiting the performance of an element through a stress daemon proves to be the major influential parameter in the problem of premature stagnation. This is due to the risk of generating incomplete structures. Users should be very careful with this parameter and either choose to limit the performance of elements with low stress levels to 0.70 and 0.90 times the performance or 0.90 for both. It is not recommended to remove the performance reduction for low-stressed elements from the routine. The strategy that works like a rejection list guides the search in a way similar to the ESO process [19].

Very slight adjustments to the highest and lowest possible reference value of an element can be used to guarantee that the process keeps trying new options even after many iterations. However, in this sensitivity analysis the stress daemon has proven to be the major influential parameter on stagnation behaviour, not the reference value reaching zero or one prematurely. If not really necessary, it is recommended not to use this function. Otherwise, a minimum reference value of 0.005 and a maximum reference value of 0.995 can be considered.

The set of parameters recommended in this paragraph is specific for this problem. A different problem might lead to a different set of optimal parameters. In case a parameter set that is used on a problem does not lead to satisfying results, the results on this sensitivity analysis can be used to track down the problem and can be a guideline in adjusting the set of parameters such that the runs become more satisfying.

6.3 Performance sensitivity

The previous paragraph focused on the determination of a good set of parameters that enables the user of the algorithm to find the best performing structure for the analysed problem. Parallel to this issue it is important to determine the performance function that describes the problem best. An arguably objective way of defining the performance of a structure is to estimate its economical costs. The costs of a concrete structure are highly determined by the volume, mould and pre-stressing. These parameters are part of the performance function as proposed for the algorithm.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1}{C_V V + C_S S + C_P P V_p + C_H H} & else \end{cases}$$

Where:

$perf_j$ = Performance of structure j.
V = Volume of the structure (m^3).
S = Surface area of the structure (m^2).
P = Pre-stress level (-).
V_p = Volume of pre-stress (m^3).
H = Level of matching the holes criterion.
C_V, C_S, C_P and C_H are constants.

This performance function calculates a rough estimation for the cost of the structure if it fits the boundary conditions. The performance is set to 1/costs. Optimising this performance means minimising the costs. Additionally, it was decided

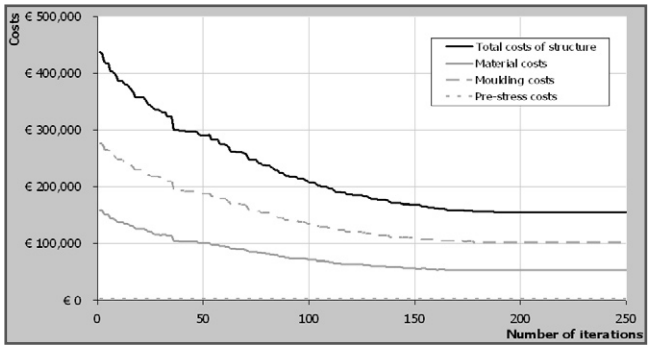


Figure 6.49: Estimated costs over iterations for best solution, performance set 1.

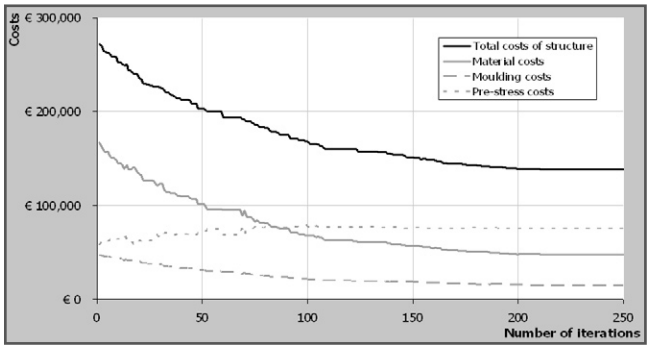


Figure 6.50: Estimated costs over iterations for best solution, performance set 2.

to add the functional criterion of holes in the performance function by assigning an extra virtual cost to structures that do not have these holes.

These parameters do not tell the whole story. Things that are not taken into account but do influence the cost of a concrete structure are, among others:

- Connections.
- Shape of the mould.
- Manufacturing considerations.

These parameters are not taken into account in the algorithm, for they are very difficult to measure. The performance function that is used is therefore very limited. In order to make the best out of the available performance function, the constants that define the influence of the parameters are to be analysed. This is done by testing the three sets of parameters: the original parameter set as used up so far and two sets that use reconsidered parameters. The influence of the parameters in the performance value will be clarified as well as the influence on the resulting structure. The runs discussed in this paragraph are the best out of four equal runs. This way, the influence of random fluctuations is minimised.

6.3.1 Defining and testing the performance

Set 1

During the testing of the algorithm up so far, the numbers that were derived in Paragraph 4.4.2 were assumed.

- Cost of material: € 2000 per m^3.
- Cost of mould: € 300 per m^2.
- Cost of pre-stress: € 500 per m^3.

These values are implemented in a run on the Omnisport case. Rather than presenting the performance, which equals 1/costs, it is in this case better to present the estimated costs, as is done in Figure 6.49.

In the graph one can observe that the estimated costs for the mould are double the estimated costs for the material. Besides, the pre-stress hardly has in influence.

Also, one can observe that the costs for moulding and the material costs are linked to a high level. More material, at the end of the day, most often means more moulding. Also, the rough representation of the mesh during the optimisation process has an influence. This can be explained with the help of Figure 5.8. The major part of the estimated moulding surface is the bottom of the mould, and is directly linked to the choice of the ants to put material or leave a void in the elements.

Any engineer can see that the relative influence of the costs is not correct. The moulding gets assigned a cost that is too high while the cost for pre-stressing is too low. The cost of the material is assumed to be correct, because this is the one value that is confirmed by experts.

Set 2

For this set, the following numbers are assumed.

- Cost of material: € 2000 per m^3.
- Cost of mould: € 50 per m^2.
- Cost of pre-stress: € 20000 per m^3.

The price of a cubic metre of UHPC is kept the same when compared to set 1. The price of the mould is now assumed to be € 50 per m^2, which is estimated by common sense. The price used in set 1 was concluded to be too high. A price of € 50 for a square metre of moulding seems correct, even though it is an intuitive value. The price of pre-stressing one cubic metre of UHPC is much higher than in set 1. Again the value assumed is based on common sense. The current value means pre-stressing an element of 250*250*5000 mm to 75 MPa would cost € 10,000.

In Figure 6.50 the estimated costs for the three parameters are shown. Now, the moulding costs are about a third of the material costs which seems plausible. The pre-stressing costs are much higher, around 150% of the material costs. This seems a bit much. Intuitively it seems not right that pre-stress costs more than the already expensive material UHPC, particularly when looking at the considered structure. As will be shown later on, the resulting structure is truss-like. Only a few of the elements need pre-stressing. It again needs to be noted that this is all, except for the material costs, merely intuitive.

In the plot a clear trade-off between material and pre-stressing can be seen. The decrease in material and moulding costs is traded for a small increase in pre-stressing costs. This shows the multi-objective optimising behaviour of the proposed algorithm.

Set 3

For this set, the following numbers are assumed.

- Cost of material: € 2000 per m^3.
- Cost of mould: € 50 per m^2.
- Cost of pre-stress: € 10000 per m^3.

This set of constants is quite similar to set 2. The only difference is the costs for pre-stressing. The constant used in this set means that pre-stressing an element of 250*250*5000

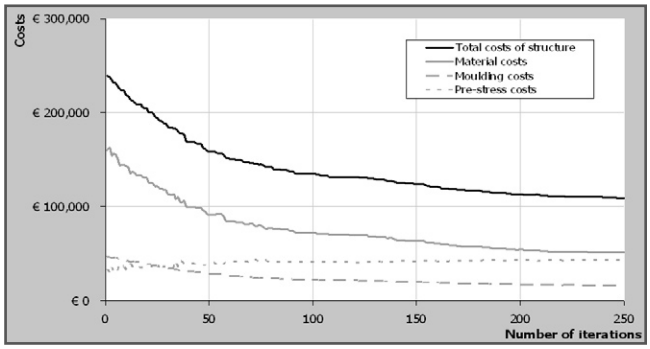


Figure 6.51: Estimated costs over iterations for best solution, performance set 3.

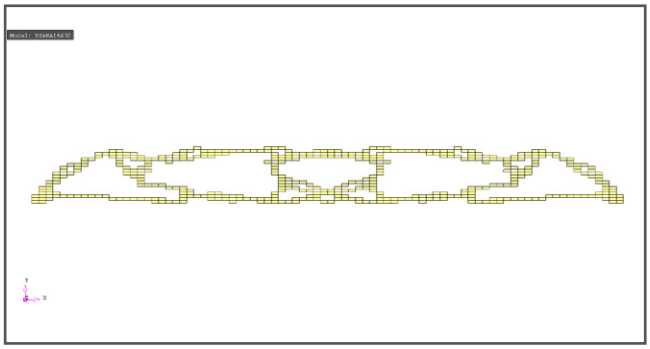


Figure 6.52: Best structure found in four runs with performance set 1. Picture mirrored along the line of symmetry.

mm to 75 MPa would cost € 5,000.

Figure 6.51 shows the estimated cost of the structure and the influence of the three parameters in this set. The results seem plausible; the cost for moulding is around one third of the material costs and the pre-stressing costs about as much as the material. Just as in the runs with set 2, the multi-objective optimising behaviour is clear in the plot of the costs. A small increase in pre-stressing costs is allowed because the costs for material and moulding can then be reduced.

6.3.2 Influence of the performance function

In Figure 6.52, Figure 6.53 and Figure 6.54 the best structures resulting from the runs with the parameter sets are depicted. Even though the structures are not similar, the structures are not very different in terms of topology either. It is concluded that the influence of the performance function on the structure that is found to be best performing is not very big if the parameters are altered in the way that has just been done. This is mainly due to the fact that the costs for material and moulding are highly linked. Together they have a considerable influence on the performance of a structure. Even though pre-stressing can be very expensive, tension forces need to be taken up in the structure and pre-stressing is therefore inevitable.

Holes constant

The influence of the holes constant has not been tested in the performance sensitivity analysis. In the problem considered, holes emerge naturally in all the runs. The holes criterion is therefore always fulfilled. In the case sensitivity analysis in Paragraph 6.4 the influence of the holes criterion will be tested by applying the algorithm to a structure in which holes do not naturally emerge.

6.3.3 Choice of performance function

In the algorithm, the performance function roughly estimates the costs of the structure. It takes into account the following parameters.

- Material costs.
- Moulding costs.
- Pre-stressing costs.
- Penalty for not matching holes criterion.

It is assumed that the parameters as tested in set 3 give a sufficiently accurate estimation of these costs.

- Cost of material: € 2000 per m^3.
- Cost of mould: € 50 per m^2.
- Cost of pre-stress: € 10000 per m^3.

These parameters cannot be validated at this point. If their actual value is slightly different, that is not a problem, because the topology that follows from the routine is not very sensitive to these parameters. In order to keep the user from having the impression that the costs of the structure are accurately estimated, the performance is calculated as 1/costs, where costs are estimated in millions of euros. In this way the user will not *read* the performance as a cost estimation. The user will see the performance grow rather than the costs diminish over the iterations. For example, the user will only see the value 10 for a structure that is estimated to cost around € 100,000.

The virtual costs for not matching the holes criterion are kept to € 20,000. The influence of this holes criterion does most probably not have an influence on the Omnisport runs, because holes are likely to emerge naturally. However, the influence of this criterion will be tested in the next paragraph.

The performance function now is defined as the following.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

Where:

$perf_j$ = Performance of structure j.
V = Volume of the structure (m^3).
S = Surface area of the structure (m^2).
P = Pre-stress level (-).
V_p = Volume of pre-stress (m^3).
H = Level of matching the holes criterion (-).

6.4 Case sensitivity

The proposed algorithm has been validated and its sensitivity to the parameters and the performance function have been determined. Another question one can ask is how the algorithm performs for different

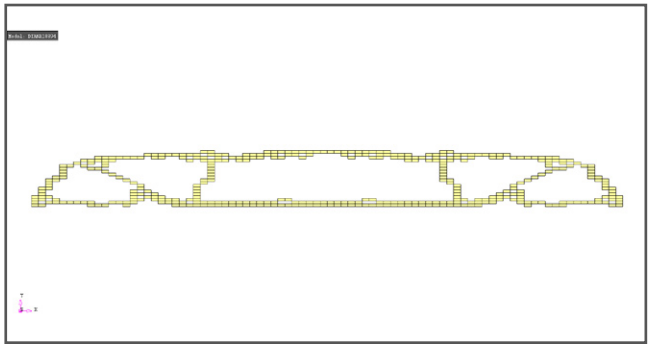


Figure 6.53: Best structure found in four runs with performance set 2. Picture mirrored along the line of symmetry.

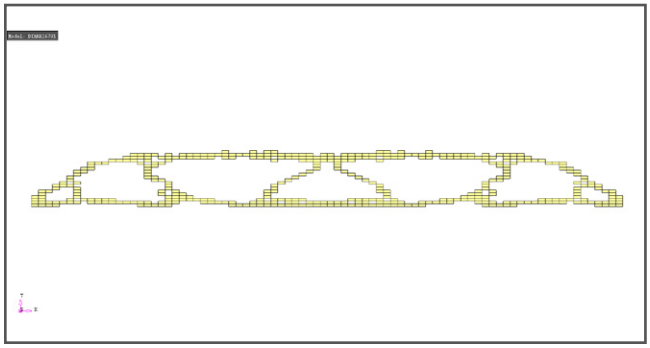


Figure 6.54: Best structure found in four runs with performance set 3. Picture mirrored along the line of symmetry.

boundary conditions. In the previous paragraphs, several structures have already been tested; a coarse approximation of the Michell structure, a fine approximation of the Michell structure and a coarse approximation of the Omnisport design case. In this paragraph, more tests will be done on the Omnisport case to see how the algorithm performs under several conditions.

The UHPC data and load cases are refined before the tests have been performed. This refined data will be discussed in Chapter 7, because that chapter focuses on the design of the Omnisport roof and the runs that were used for this design. This paragraph merely shows the behaviour of the algorithm for several different cases, for which the exact UHPC and load data is not of major importance.

6.4.1 Coarse model

Input

The boundary conditions for this run are given in Figure 6.55. Because of symmetry reasons, only half the space available for structural elements is used in the algorithm. This space, a 52.5 by 10 metre space, is meshed to 42 by 20 elements. Elements have a 0.10 metre thickness. The material properties of ultra-high performance concrete are used and the performance function as defined in Paragraph 6.3.3 is implemented. Full input as well as output details are given in Appendix D.4.1.

Optimisation process and output

To diminish the influence of random fluctuations, six parallel runs have been performed on this problem. Each run shows a similar optimising behaviour. All the solutions are truss-like, with very similar upper and lower chords and slightly different diagonals. An image of the best structure found in the runs is given in Figure 6.56. The performance of this structure is 7.8.

The chords are thicker in the middle of the span. Their size depends on the bending moment, which is larger at that point. The diagonals in the structure only appear at those places where shear forces occur; near the sides and not so much in the middle. The algorithm only analysed the linear-elastic behaviour of the structure and did not take into account stability. There was, in other words, no incentive to make diagonals which logically results in the program not making them.

6.4.2 Fine model

Input

Using the same boundary conditions as in the coarse model but increasing the mesh resolution, new runs are performed. Again only half the structure is analysed because of symmetry reasons. The space is now meshed to 105 by 50 elements. Because the search space is a lot bigger than in the case of the coarse model, the number of ants is doubled in order to be able to search the bigger space well. Element and material data is similar to the coarse run and is, together with the output details, given in Appendix D.4.2.

Optimisation process and output

Just as with the coarse model, six runs have been performed on this design problem. Finding an optimum is now a bit more difficult. For this reason the optimisation chart has been given in Figure 6.57. The jumps in performance in

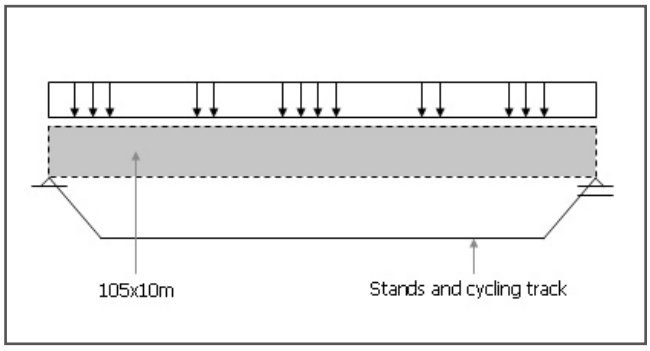


Figure 6.55: Boundary conditions for the Omnisport case.

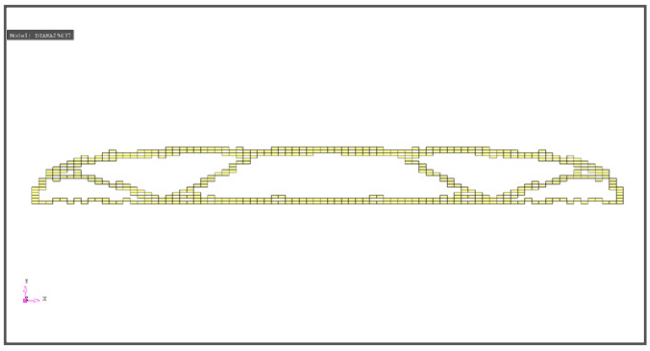


Figure 6.56: Best solution for coarse approximation. Picture mirrored along the line of symmetry.

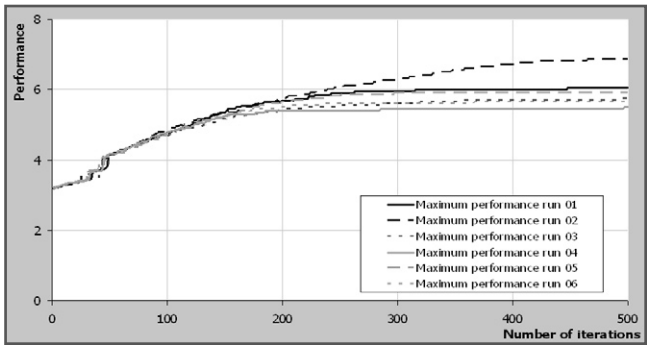


Figure 6.57: Performance chart for runs on fine model.

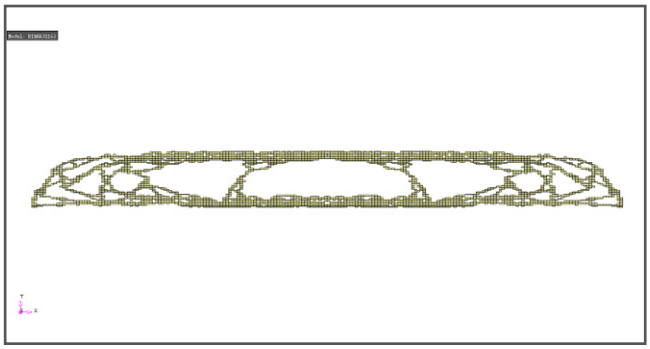


Figure 6.58: Best solution for fine approximation. Picture mirrored along the line of symmetry.

between the 25th and 50th iteration are due to the holes that develop in the solutions. The fact that the holes have the same size, but comprise of more elements that need to be left empty makes it more difficult for the algorithm to implement these holes.

At the end of the iteration the difference between the best structure found in the different runs is considerable. Apparently the bigger search space made optimisation difficult and more sensitive to the influence of randomness.

The best structure that is found in the runs is given in Figure 6.58. Its performance is 6.8. Again, diagonals develop where the shear force is relatively large: near the sides. The varying thickness of the upper and lower chord can be seen even more explicitly than in the coarse approximation. The high resolution seems to have an influence on the diagonals near the sides; some are connected eccentrically. The almost organic shape has a slightly lower performance than the shape derived from the coarse model. In terms of topology, the two runs yield fairly similar results. The higher resolution has an influence on the results. It makes the search space bigger and therefore it is more difficult to find an optimal structure, which results in a performance value that is slightly lower than in the coarse run.

Analysis of the stagnating runs

A specific problem for these runs is visually explained in Figure 6.59. Some runs stagnate at a result as shown in the top-left corner of the picture. Further optimisation generates solutions that do not have the parts circled in the mid-left image. Because of this, the part circled in the bottom-left corner are not supported anymore. The connectivity analysis is not powerful enough to delete this whole bunch of elements. Because the load is applied to the top of the mesh (top-right image), this part collapses onto the structure yielding big displacements in this part of the structure. The performance function now assigns a zero performance to this solution because of these displacements. This problem would not occur if the load would be applied to the bottom of the mesh, as shown in the bottom-right image.

6.4.3 Complete model

Input

In order to see what happens if the full problem is analysed and not only half the structure, the complete model is implemented in the algorithm. The same boundary conditions as for the coarse and fine symmetrical approach apply. Now, the full space is meshed to 84 by 20 elements.

Optimisation process and output

Four runs have been performed on this problem. As could have been expected, each ant run takes about double the time of the coarse model. Because of the bigger search space double the amount of ants was used in comparison to the coarse runs, and the run times therefore go up with a factor of around four.

The best result found in the runs is given in Figure 6.60. The result is a bit more messy than the result on half the model. Also, the performance is a bit lower: 7.1. As can be seen in Appendix D.4.3, the performance of the best solutions found in the different runs is almost equal.

This run has shown that, just as with the fine model, it is more difficult to find a clear optimum when the search space is increased. In terms of topology the structure that is found resembles the structure found with the coarse approximation which means the algorithm is robust for these cases.

6.4.4 Load on the bottom side of the mesh

Input

Because of the findings on the fine model with respect to the application of the load on the top side of the mesh, runs were set up with the load on the bottom side of the mesh to check how this influences the result. The boundary conditions are shown in Figure 6.61. Except for the different location of the load, run data was kept similar to the coarse run.

Optimisation process and output

Full run data is given in Appendix D.4.4. The performance development of the different run is almost equal. The best solution is depicted in Figure 6.62 and has a performance of 7.4.

Two interesting things about the runs are highlighted. Firstly, the diagonals are now pointing in the other direction and secondly, the solution performs slightly worse than the coarse run with the load on the top side of the mesh. The different orientation of the diagonals can be explained easily. The load is applied to one of the sides of the mesh. Besides the normal forces that occur in a truss, the chord that is located on the loaded side of the mesh is also bent by the load. To prevent very big bending moments and big displacements, this chord is supported in regular intervals by the diagonals.

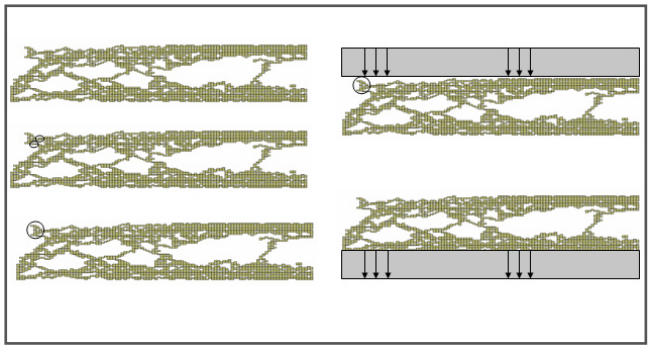


Figure 6.59: Problems with the fine approximation.

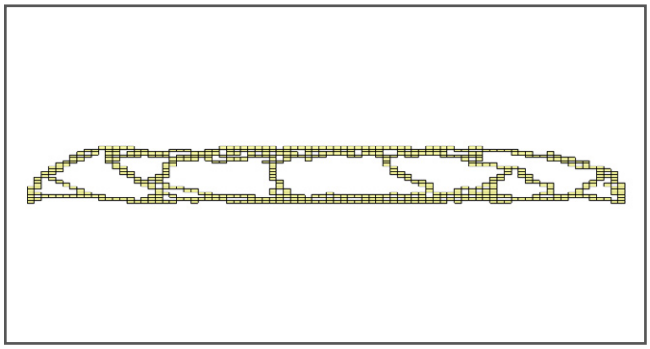


Figure 6.60: Best solution for the complete problem.

This is the reason why the main diagonal now supports the bottom chord while it supported the upper chord in the runs where the load was applied to the top of the mesh. Following from this, the main diagonal now has a tensile force. This means the diagonal has to be pre-stressed, which results in higher costs of the structure and thus a lower performance.

The reason why this run was performed was because of the problems spotted in the fine runs. There, material would not be supported and collapse onto the structure yielding big displacement. It was feared that this phenomenon would lead to the algorithm finding different than optimal topologies because of this problem. The results from this run show that merely the orientation of the diagonal differs, for reasons stated above. As explained in Paragraph 5.3.6 it was chosen to estimate the dead load of the structure and not calculate is as the algorithm runs. One of the disadvantages of this approach is that diagonals that are needed to support the dead load of the chords will not develop in the algorithm. The load that is applied externally to the structure is applied to the roof of it on the top of the UHPC element that supports it. For this reason it is most accurate to apply the load to the top of the structure. The user has to be aware of the fact that dead load is accounted for in this load and that extra elements might be necessary to support the dead load of the individual elements.

6.4.5 Up- and downward load cases

Paragraph 5.3.6 focused on the load cases that can be implemented in the algorithm. Without explicitly noticing, the Omnisport cases that are considered all use four load cases: two serviceability limit states and two ultimate limit states. All these loads, however, were pointing in the same direction up so far which means scaling the stresses and displacements is the only necessary thing to do to calculate values for the different load cases.

Input

A very interesting case arises when the load cases do not all point in the same direction. Such a case, as depicted in Figure 6.63, can occur if a light structure is constructed; wind suction will create an upward load case while snow loading is pointed downward. Meshing and material properties are similar to those used in the coarse runs. During the first runs on the problem, many runs stagnated prematurely. With the knowledge generated by the parameter sensitivity analysis it was decided to diminish the influence of the stress daemon. This proved a good adjustment. Further run data is given in Appendix D.4.5.

Optimisation process and output

The performance development chart in Appendix D.4.5 shows that there is a considerable difference in the performance of the structures generated in the different runs. Figure 6.64 shows the best structure found in the different runs.

The difficulties that the algorithm finds when optimising the structure are mainly because it is difficult to keep the maximum compressive and tensile stress in the members in the elastic range. If tensile stresses occur, pre-stress is necessary, which lowers the compressive capacity.

The diagonals in this run do not have a clear orientation and the performance of the resulting structure is 7.4; slightly but not dramatically lower than the performance found for the case with loading in a single direction. Especially the top

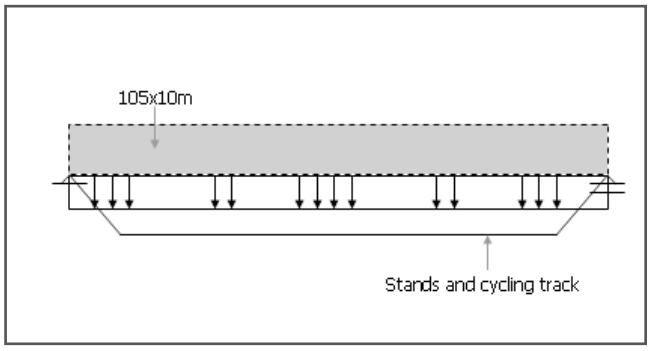


Figure 6.61: Boundary conditions for run with load on the bottom of the mesh.

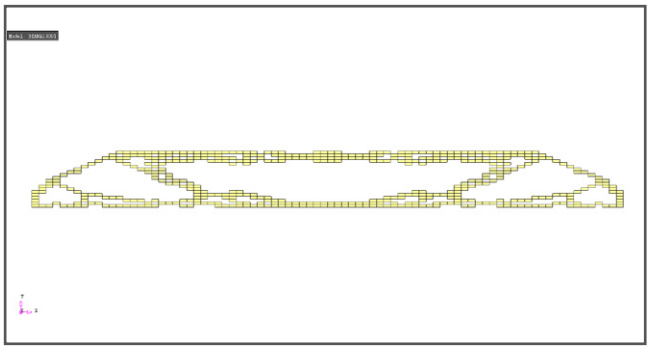


Figure 6.62: Best solution for run with load on the bottom of the mesh. Picture mirrored along the line of symmetry.

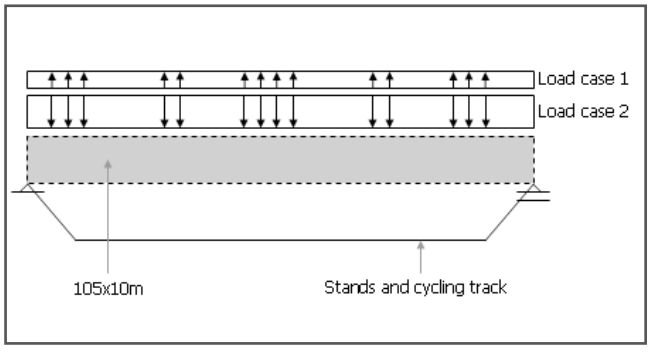


Figure 6.63: Boundary conditions for run with up- and downward load cases.

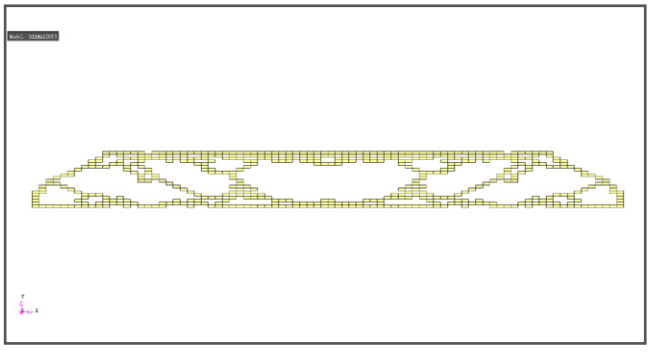


Figure 6.64: Best solution for run with up- and downward load cases. Picture mirrored along the line of symmetry.

chord is bigger and more often supported than in the case with one-directional load case; a logical result because this chord is loaded is directly loaded by the up- and downward forces. The fact that, even though the optimisation process was more difficult, a result evolves with a performance that is just 5% lower than the result for the one-directional load case is satisfying.

6.4.6 Pinned model

In the test runs up so far holes would naturally emerge in the structure because of the nature of the design problem; a truss-like structure would emerge and such a topology contains holes by nature. Only the fine approximation did not generate structures with holes right from the start.

Input

To see what the effect of the holes parameter in the performance function actually is, four runs have been performed on the problem represented in Figure 6.65. The structure is now pinned in both corners. This problem was analysed earlier in Paragraph 6.1.3. In that analysis, an arch evolved. Because of symmetry only half the structure is analysed. All parameters except for the supports are equal to those in the runs with the coarse mesh. Detailed run data is given in Appendix D.4.6.

Optimisation process and output

The generated structures contain holes right from the first iteration onwards. The best result obtained in the results is shown in Figure 6.66. The structure is cheap because only little material is needed. Its performance is 20.8. Interestingly, more than just the four holes that were asked for have developed during the process. This is explained with Figure 6.67.

During the process, the circled holes develop in the structure. Because of these holes, the force needs to flow around them. The lines represent the force flow in the structure. Now the elements in between these lines are not of major importance anymore; they are not stressed to a high level. Because of this, the holes grow and new holes develop in these areas.

6.4.7 Ordinary concrete

Input

Runs have been performed on the same problem as in the coarse runs, but now with ordinary concrete that has a characteristic compressive strength of 65 MPa. Calculations on the stress-strain diagram that is used in the calculations is given in Appendix B.2. In the same appendix the results of an iterative estimation of the self weight are given. Complete run data is given in Appendix D.4.7. Because the concrete is less strong, more material is needed to cope with the loading. This results in a bigger dead load that again asks for more material, and so on. To be able to cope with the big sizes of concrete, the thickness of the elements in the mesh is increased from 0.1 to 1.0 metres. Because this kind of concrete is cheaper, the parameter for the costs of concrete in the costs estimation is lowered to € 300 per cubic metre. This is a rough assumption based on the rule of thumb that one cubic metre of concrete including moulding costs about € 350 once fully in place.

Optimisation process and output

The optimisation in the four runs is very alike. The best

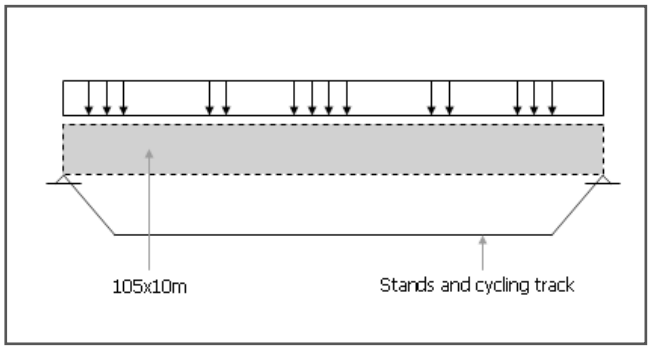


Figure 6.65: Boundary conditions for run with non-trivial holes.

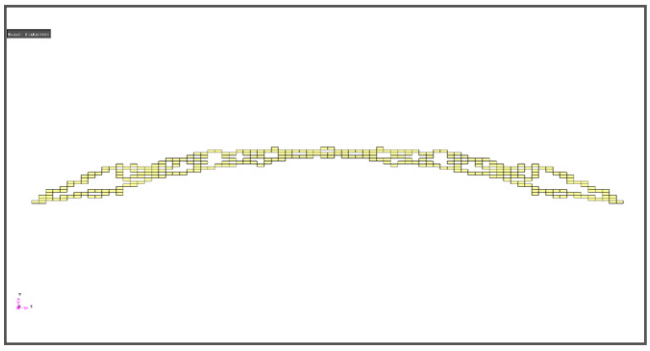


Figure 6.66: Best solution for run where holes are not trivial. Picture mirrored along the line of symmetry.

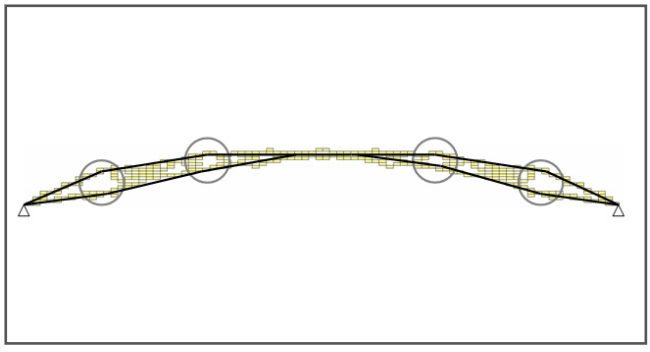


Figure 6.67: Analysis of the extra holes evolving in the structure.

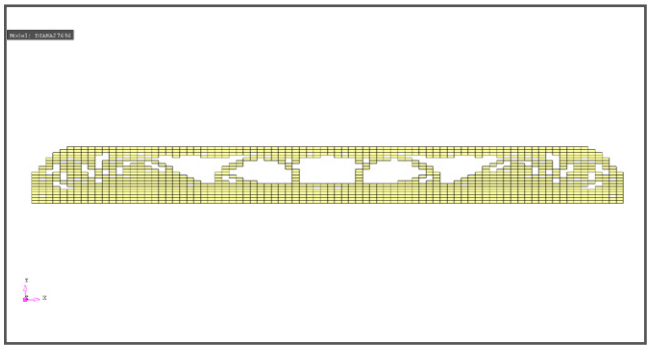


Figure 6.68: Best solution for run with ordinary concrete properties. Picture mirrored along the line of symmetry.

resulting structure is given in Figure 6.68. The performance of this structure is only 0.95. This is due to the large amount of material used in the structure. It should be noted, though, that the pre-stressing has a major influence on the costs. The costs of pre-stressing are calculated based on the costs of fully pre-stressing one cubic metre of UHPC. Because ordinary concrete does not have to be pre-stressed to the same stress as UHPC, the costs for pre-stressing will be lower than estimated. On the whole, the performance function is not well developed for this run. This is not truly a problem, because the run was performed merely to see if a structure in ordinary concrete would be possible.

It can be concluded that a structure in ordinary concrete is possible, but not very logical. A lot of material is needed which results in a huge and expensive structure. The span is difficult to cover in ordinary concrete because its dead load is a lot bigger than if a different kind of material would be used.

6.4.8 Enlarged search space

Up so far the algorithm has been used for problems that are still fairly trivial; Michell structures, simply supported girders and arches. To see what solutions the algorithm comes up with for less trivial problems, the Omnisport case is reconsidered.

Input

The space available for the placing of structural elements is enlarged for this case to the space showed in Figure 6.69. The less trivial search space might lead to the algorithm developing solutions that an engineer or designer would not directly think of. The size of the elements in the mesh is kept the same as in the coarse runs, but the bigger search space leads to a mesh consisting of 4406 elements. Loads are applied to the top of the mesh and UHPC properties are implemented.

Optimisation process and output

The run data and performance charts, shown in Appendix D.4.8, show very similar optimising behaviour for the four performed runs. Indeed, all structures are fairly similar but not quite as expected. Beforehand, it was expected that the algorithm would come up with a cable stayed structure. In fact it does not. The best structure found in the runs is shown in Figure 6.70 and has a relatively poor performance of 5.4.

The best performing structure does not use the extra support at all. It follows the boundaries of the mesh in the rest of the space that is available for the placing of structural elements. Very long and slender members develop. The resulting structure does not seem very robust or practical.

6.4.9 Extra support

Input

An extra column is placed as another adjustment to the original boundary conditions to give the algorithm another chance to show its capabilities of finding optimal structures for non-trivial problems. The boundary conditions used are shown in Figure 6.71 and described in more detail in Appendix D.4.9.

Optimisation process and output

Four runs are performed on this problem. The best solution found in the runs is shown in Figure 6.72, has a very

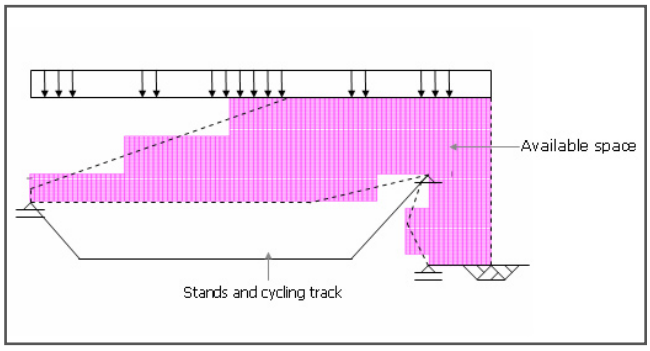


Figure 6.69: Boundary conditions for enlarged search space.

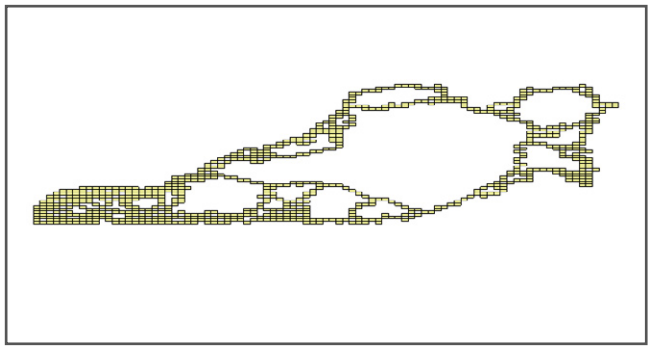


Figure 6.70: Best solution for run with enlarged search space.

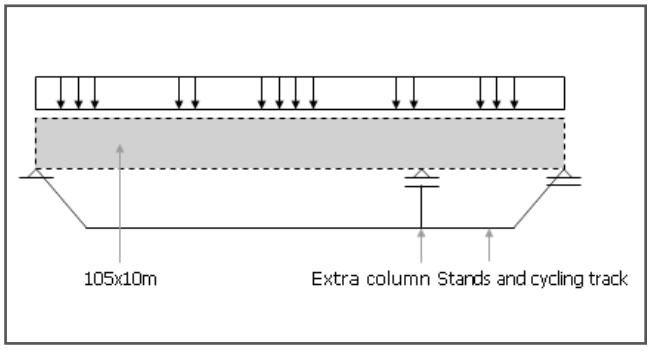


Figure 6.71: Boundary conditions with extra support.

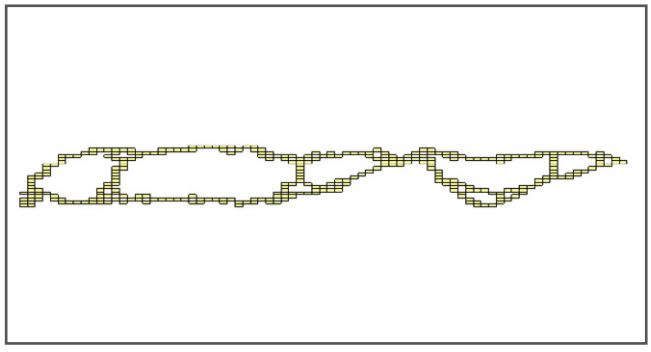


Figure 6.72: Best solution for run with extra support.

high performance of 13.7 and is interesting to analyse.

Analysis of the resulting structure

Only two of the three supports are used by the structure, that more or less consists of two parts. At the right of the structure a cantilever evolves. The loading on top of this cantilever causes a rotation around the middle support and a resulting upward force on the very small connection of the two parts of the structure, as is shown in Figure 6.73. This upward force supports the left part of the structure, that can now be constructed with very slender members.

The algorithm found an optimal structure for this specific case that has a very high performance, but the structure is not very robust. If any other load case would apply to the structure, the very unstable equilibrium is disturbed and the structure will collapse as shown in Figure 6.74. For this reason the structure cannot be implemented in practice.

From the results of this case it can be concluded that the implementation of the problem in the algorithm is of major importance; the optimal structure that the algorithm finds is specific to the implemented case.

6.5 Conclusions

In this chapter the validation, parameter sensitivity, performance sensitivity and case sensitivity studies of the proposed algorithm are reported.

6.5.1 Validation

The validation studies on the Michell truss problem have shown that the algorithm is capable of finding the optimal topology for a structural design problem. These studies have also shown that using as few elements as possible makes the results more reliable. Increasing the resolution of the mesh brings along the risk of not finding an optimum because of the increase in size of the search space. Further analysis has shown that the calculations done in DIANA are correct and that peak stresses that are visible in the DIANA post-processor but that are not very likely in the final design are not taken into account by the algorithm. Also, analysis has shown that the material representing the voids behaves satisfactory.

6.5.2 Parameter sensitivity

A study on the sensitivity of the optimisation process to the parameters that are used in the algorithm has given an idea of an optimal parameter set. The dependency of the optimisation routine on the parameters has been clarified. The optimal set of parameters is, however, case-specific. The user is advised to start with using the proposed set of parameters and use the clarified dependencies as a guideline to adjust the parameter sets if satisfactory results are not obtained.

6.5.3 Performance sensitivity

Multi-objective optimisation is possible due to the implementation of a multi-objective performance function in the algorithm. Many aspects, both structural and non-structural, can be used as an optimisation objective. Studying various performance functions and clarifying the influence of the different parameters in the performance function has shown that the algorithm is capable of performing a trade-off between different objectives even when the influence of these objectives on the performance of the solution is contradictory.

The insight is generated that for the case to which the algorithm is applied, the constants in the performance function do not have a large influence on the structural topology that the algorithm proposes. The performance function that is used on the problem takes the estimated costs as a guideline. The estimation of these costs is very rough, based on assumptions and is by no means claimed to be accurate.

6.5.4 Case sensitivity

Applying the algorithm to many cases with different sizes, resolutions, supports, loads and material properties has shown that the algorithm is widely applicable and robust. For each problem, a structure is found that makes sense for the specific problem. However, also the drawbacks of the algorithm have become clear. Especially for cases that are not statically determined, the tool can come up with solutions that might be optimal for the problem implemented in the algorithm, but are not applicable to the actual design case.

When applying the algorithm to a case, the user has to be aware that all the inaccuracies that are made during the implementation of the case in the algorithm will influence the design that the routine comes up with.

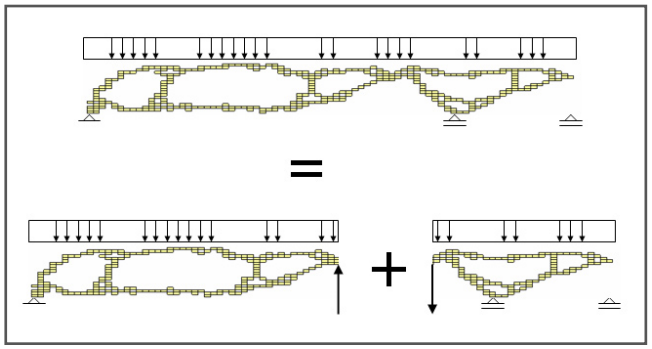


Figure 6.73: Mechanical analysis of the solution from Figure 6.72.

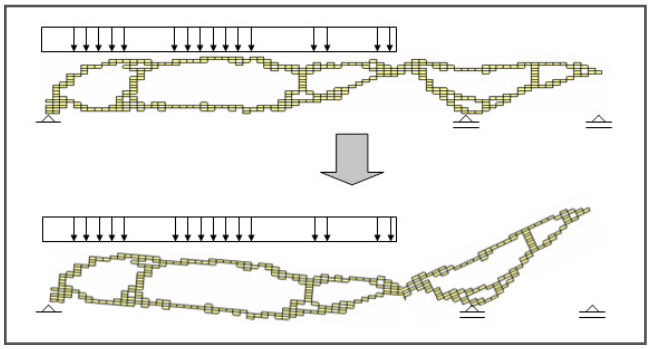


Figure 6.74: Collapse of the structure due to instability.

6.5.5 Function of the tool

The proposed algorithm is aimed at developing optimal topologies for structural design problems. Indeed, the algorithm is capable of doing so. Testing has shown that the solutions generated by the algorithm cannot be directly implemented in any design. This is due to the restrictions of the tool; only boundary conditions and considerations that can be implemented in the tool can be optimised for.

The algorithm can be a useful topology optimisation tool that generates several solutions that inform the designer about the optimum solution for the case as specified. The designer needs to interpret the solutions; he has to distil members and connections from the plots and needs to develop the structure further according to the boundary conditions and performance parameters that have not been taken into account by the algorithm.

CHAPTER 7 DESIGNING THE OMNISPORT ROOF

The objective of this thesis is to create and evaluate a structural design algorithm, based on the concept of the ant system, which structurally designs a roof for the Omnisport building in Amersfoort, the Netherlands, in ultra-high performance concrete. This chapter elaborates on the actual design of the Omnisport roof inspired by the results of the structural design algorithm.

Before applying the algorithm to the case, the exact model that is implemented in the algorithm is developed. Then the output of the algorithm is analysed and a choice for a proposed topology is made. A preliminary design is generated based on this topology.

7.1 Algorithm input

The algorithm input consists of a mesh, the support conditions, the loads applied to the problem, the properties of the considered material, a performance function and a parameter set for the optimisation process. Complete and detailed run data can be found in Appendix D.4.1 and D.5.

7.1.1 Mesh

The optimisation problem

The hall has a 120 by 105 metre floor plan as shown in Appendix A. During the development of the algorithm awareness has risen that the runs that are performed with the proposed algorithm take a long time, depending on the size of the problem. The proposed algorithm can be applied to three dimensional problems. Because of symmetry, analysing the structure in 3D would mean that only a quarter of the structure has to be analysed. If one would assume a maximum allowable height for the structure of 10 metres this would result in a 60 by 52.5 by 10 metre space. Approximating this space very coarsely with elements of 1 by 1 by 1 metre yields a mesh of around 31,200 elements. The estimated run time based on the rough approximation derived in Paragraph 6.2.7 for optimising the 31,200 element mesh is 13,628,160 seconds, which is equal to 158 days. Even though the approximation is developed for 2D structures and is not very well validated, it gives a feel of the order of magnitude of the run time needed. Quick optimisation is simply not possible with the current algorithm for a structure of this scale in 3D.

Because of this finding a two dimensional cross section of the structure is analysed. In the structure as designed by the architects and engineers on the project, 2D steel trusses support the roof along the roof span. In order to be able to truly compare the solution found by the algorithm with the applied solution, it is chosen to take the 2D cross section as

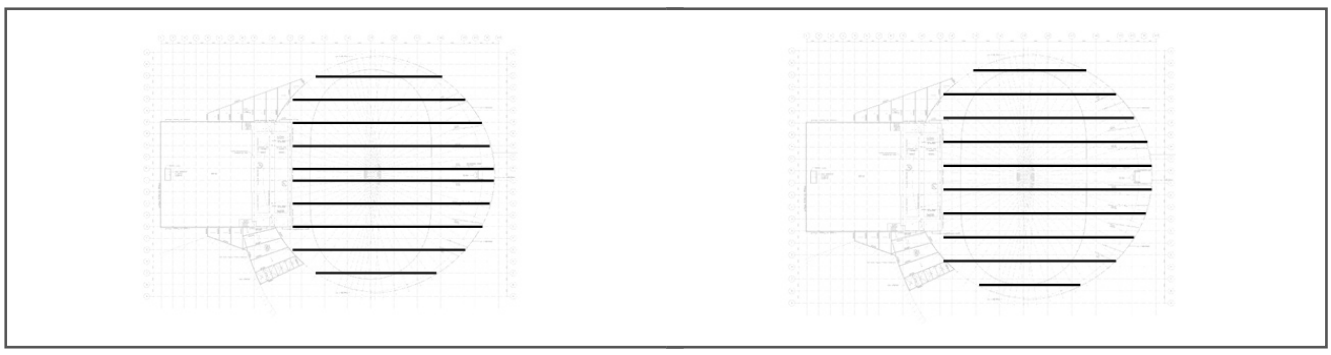


Figure 7.1: Current and proposed layout of the structural elements.

a starting point. Two dimensional load-bearing elements are designed in UHPC. The design will be based on the findings of the algorithm and will be such that it can directly replace the steel trusses in the current design.

The cross sectional drawings of the Omnisport design in Appendix A show the size, placement and orientation of the steel trusses. The trusses are mostly placed at a 12 metre centre to centre distance. This is not the case for the central trusses; they are placed 6 metres from each other. This is done because it matches the grid layout better, because these two trusses span the longest distance and because a walking bridge for maintenance is placed in between. This walking bridge gives gives an extra load to the structure and the visual perception that the bridge and the two trusses form one element. The trusses have a depth of 5 metres.

For the optimisation of the 2D cross section as will be performed with the algorithm, it is assumed that a depth of the elements of 10 metres is possible. This will give more design options. The design in UHPC is supposed to be a very clean and neat structure, so that the bigger elements will not become of visual hindrance. The centre to centre spacing will be made a uniform 12 metres. Now when the depth of the structural elements is bigger, the maintenance bridge will be too small to give the perception that it forms one element with the structural elements beside it. Going to a 12 metre centre to centre spacing will increase the load on the structural elements, but will give a more uniform visual perception from the inside.

Figure 7.1 shows the current and the proposed layout of structural elements. The same figure also clarifies why the choice for trusses or other structural elements that span in one direction is logical. The roof plan is not circular but elliptical. Moreover, one side of the ellipse is cut by the square building next to the cycling hall. The resulting cut ellipse has a clear orientation. One span in longer than the other. It is a logical choice to place structural elements in the short span of the roof and secondary girders or roof plates in between these elements.

Design space

Figure 7.2 shows the space available for the load-bearing structure for the element with the largest span as a result of the decisions made above. The boundary conditions are kept as identical as possible to the boundary conditions that apply to the actual Omnisport hall roof structure. This means that the roof is simply supported. On the bottom left side of the design space, a hinge-like support can be constructed that takes horizontal and vertical loading but allows for rotations. On the right side the structure is supported by a column which will result in a rolling support of the structural element.

The design space is slightly inclined. In terms of structural behaviour, this inclination does not have an effect on the structural element, which is still simply supported. In the mesh that is developed, the design space that is used does not have this inclination. This makes the meshing easier, does not have any consequences on the resulting structure and makes gaining insight in the results easier.

The resulting design space is shown in Figure 7.3. It has a width of around 105 metres, a height of 10 metres and is symmetrical.

Resulting meshes

Because of symmetry reasons, it is possible to analyse only half the structure. As is shown in Paragraph 6.2.7, the number of elements in the analysed mesh highly determines the efficiency of the optimisation process. For this reason it is decided to perform optimisation runs on only half the mesh and mirror the results along the line of symmetry. The mesh is restrained horizontally along the line of symmetry and vertically in the bottom left corner as is depicted in Figure 7.4.

Two meshes are considered in the optimisation runs discussed in this chapter: a coarse and a fine approximation of the design space. Both meshes have been used earlier in Paragraph 6.4 when the sensitivity of the algorithm for the analysed case was determined. Now when the meshes are actually used for the generation of a design, a more extensive explanation of the meshes is given.

The coarse mesh splits half the design space, an area of 52.5 by 10 metres, into 1.25 by 0.5 metre elements. This results in a 42 by 20 element mesh as shown in Figure 7.5. The finer mesh uses 0.5 by 0.2 metre elements, resulting in a 105 by 50 element mesh as shown in Figure 7.6. The element consists of 2D plain stress elements as explained in Paragraph 5.2.1. These elements need to be assigned a thickness which determines their cross sectional surface. Especially for the coarse mesh, the risk of choosing a rather large element thickness is that the surface area of one element is so big that the stress in it will be very small and the structure will not be very efficient. Based on iterative calculations that can be found in Appendix D.5, that will be discussed in the next paragraph, it is estimated that members with a surface of around 0.1

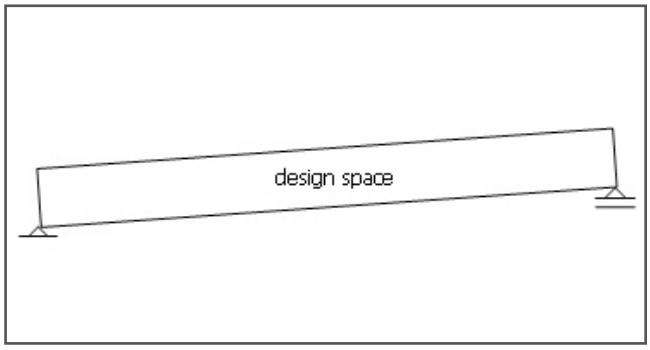


Figure 7.2: Design space and supports. Based on the drawing in Appendix A.

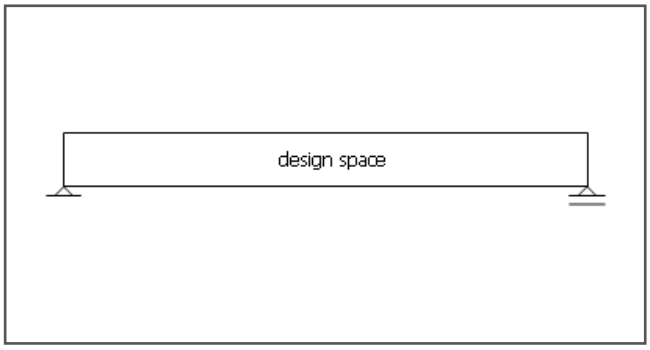


Figure 7.3: Design space and supports used in the algorithm.

m^2 will develop. In order to allow this and to leave room for members with a smaller cross section to develop, the cross section of the elements is the mesh should be around half this size: 0.05 m^2. Based on this the element thickness is chosen to be 0.1 metre.

7.1.2 Load cases

In the optimisation run for the Omnisport centre, four load cases are considered.

- Serviceability limit state (SLS) snow. This includes snow- and dead load.
- Ultimate limit state (ULS) snow. This includes factorised snow- and dead load.
- Serviceability limit state (SLS) wind. This includes upward wind- and dead load.
- Ultimate limit state (ULS) wind. This includes factorised upward wind- and dead load.

These load cases were governing during the design process of the Omnisport hall. Original calculations are given in Appendix A. As explained in Paragraph 5.3.6 the self weight of the structure is estimated a priori. This is done according to an iterative calculation process that can be found in Appendix E.1. The UHPC elements are developed such that they can directly replace the steel trusses in the current design. The loads on the newly developed load-bearing structure are assumed to be identical to the loads that were determined in the original calculations.

For the four load cases, the following loads are taken into account.

- Self weight: 0.70 – 1.00 kN/m^2
- Applied load: 0.80 kN/m^2
- Snow load: 0.56 kN/m^2
- Wind load: -0.95 kN/m^2

The self weight is set a as a range because it is not yet known at the start of the optimising process. For upward load cases the smaller number will be considered, whilst for downward load cases the larger number will be taken. The values above result in the following load combinations.

- SLS snow: *1.00+0.80+0.56* = 2.36 kN/m^2
- ULS snow: *1.2*(1.00+0.80)+1.5*0.56* = 3.00 kN/m^2
- SLS wind: *0.70+0.80-0.95* = 0.55 kN/m^2
- ULS wind: *0.9*(0.70+0.80)-1.5*0.95* = -0.08 kN/m^2

Snow load is governing for the structure. There is a possibility that a very small resulting upward load occurs in the ultimate limit state for wind. This load is almost negligible. Combined with the knowledge generated in the case sensitivity analysis, where it was found that runs for upward and downward load cases yield results that are not quite as clear as cases where only load cases in one direction are analysed, it is decided to only input downward load cases in the algorithm.

It is noted that due to a typing error in the program the optimisation routine uses a downward SLS load of 2.26 kN/m^2 instead of 2.36 kN/m^2. Unfortunately this error was not spotted during the checks on the implemented data. The SLS load has effects on the deformation of the structure. Fortunately this proved not to be governing during the runs.

After the optimisation runs are performed on the structure, the result will have to be checked on stresses and displacements. During these checks the possible upward load and the deflections will need special attention.

7.1.3 Ultra-high performance concrete

Calculations on the stress-strain diagram of the ultra-high performance concrete BSI are given in Appendix B.1. Only a portion of the linear-elastic range of the resulting stress-strain diagram is used.

Compressive stress

Stress values are governing in the ultimate limit state. For the compressive stress a value of 102 MPa is possible. However, it is wise to lower this allowable stress because of the risk of buckling. It is decided that a 25% reduction will be applied. This results in an allowable compressive stress of 76.5 MPa.

Tensile stress

The calculations show that the elastic tensile stress is 4.6 MPa. Higher tensile stresses can be allowed when pre-stress is applied. In that case it is assumed that the concrete can be pre-stressed up to its full compressive strength: 102 MPa. Due to creep, shrinkage, relaxation et cetera, losses in the pre-stressing will occur. A conservative assumption for pre-stress losses is 20% [17]. This results in an allowable tensile stress of 81.6 MPa.

7.1.4 Performance

The performance function and performance parameters have been explained earlier, and a good set of constants is

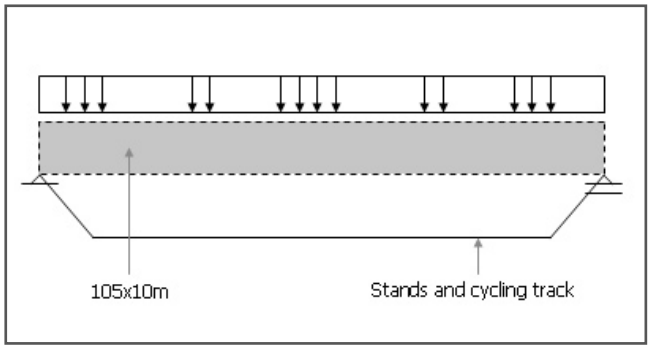


Figure 7.4: Meshed area and constraints.

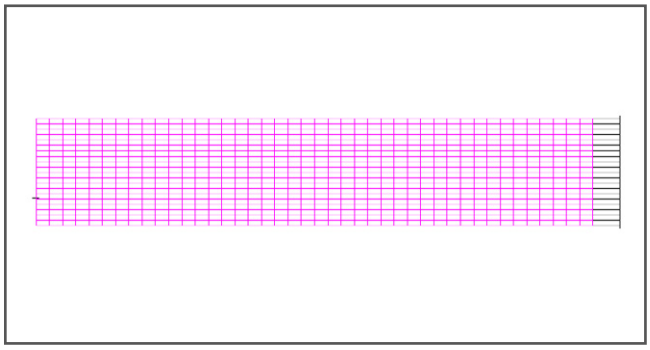


Figure 7.5: Coarse mesh.

derived in Paragraph 6.3.3. The resulting performance function is the following.

$$perf_j = \begin{cases} 1 \cdot 10^{-6} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-5} & if\ \sigma_{max} > \sigma_{allowable;max} \\ 1 \cdot 10^{-5} & if\ \sigma_{min} < \sigma_{allowable;min} \\ 2 \cdot 10^{-5} & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

Where:

$perf_j$ = Performance of structure *j*.
V = Volume of the structure (m^3).
S = Surface area of the structure (m^2).
P = Pre-stress level (-).
V_p = Volume of pre-stress (m^3).
H = Level of matching the holes criterion (-).

The maximum displacement in the structure is a boundary condition, just like the maximum and minimum stress. The stress range in an element in each direction is also limited, because allowable compressive stress goes down with the application of pre-stress. The maximum displacement is set to *span/200* = 0.5 metres. If the boundary conditions are fulfilled, the performance is calculated as a value in which volume, mould surface, pre-stressing and the matching of the holes criterion have an influence. It is assumed that the same amount of walking bridges is needed as in the original design. In the original design the walking bridges have a size of 1.0 by 2.0 metres. Four bridges are going through the trusses.

7.1.5 Parameters

The parameters used in the optimisation algorithm are based on the optimal set derived in Paragraph 6.2.8. For the fine mesh, the number of ants is doubled to 200 in order to be able to explore the bigger search space properly in every iteration.

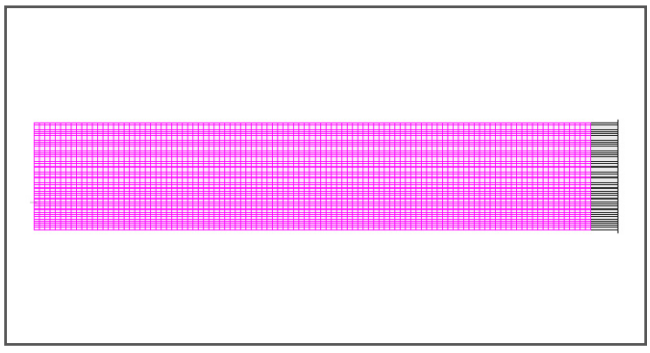


Figure 7.6: Fine mesh.

7.2 Algorithm output

7.2.1 Coarse runs

The run data and resulting structures of six coarse runs are given in Appendix D.4.1 and D.5.1. All images shown are mirrored, because only half the structure was analysed and the symmetric properties of the problem were used. All resulting structures have a similar topology; a truss with a few diagonals near the sides. In the centre of the span, no diagonals develop. The best performing structure is given in Figure 7.7.

The performance parameters of this structure are as follows.

- Volume: 26.8 m^3.
- Mould surface: 334 m^2.
- Volume of full pre-stress: 5.70 m^3.
- Holes criterion: Matched.

All this results in a performance of 7.8.

As mentioned in Paragraph 6.4.1, the diagonals in the structure only appear at those places where shear forces occur, and the size of the chords depends on the bending moment, which is bigger in the middle of the span. The lack of diagonals in the middle makes the structure instable. However, the algorithm only analysed the linear-elastic behaviour of the structure and did not take into account stability. There was, in other words, no incentive to make diagonals which logically results in the program not making them.

7.2.2 Fine runs

Run data and resulting structures on the optimisation process performed on the fine mesh is shown in Appendix D.4.2 and D.5.2. Resulting topologies are very alike. Discrepancies and problems during the optimisation process have been discussed in Paragraph 6.4.2. The best performing structure emerging from the fine runs is given in Figure 7.8.

The performance parameters of the full structure are as follows.

- Volume: 31.2 m^3.
- Mould surface: 488 m^2.
- Volume of full pre-stress: 6.00 m^3.
- Holes criterion: Matched.

All this results in a performance of 6.8.

Again the diagonals are positioned close to the sides of the structure and the upper and lower chords have a bigger cross section in the middle of the span. In the middle of the span,

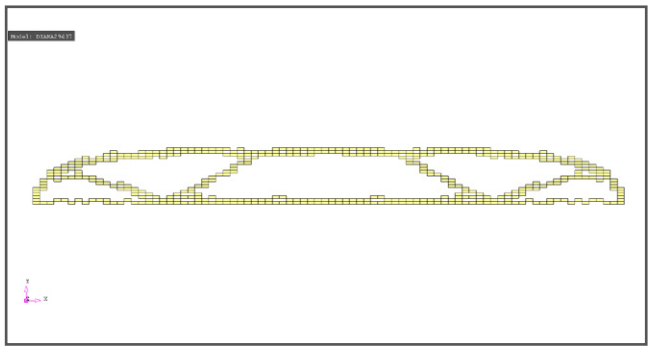


Figure 7.7: Best performing structure emerging from the coarse mesh. Picture is mirrored along the line of symmetry.

stability is not secured because of the lack of diagonals.

7.2.3 Choice of topology

The structures that emerge from the runs on the fine mesh have many very small members, especially near the sides, creating an almost organic shape that could resemble the structure of a bone.

One of the resulting topologies is chosen for further development. Based on the performance function considered in the algorithm the coarse model performs better. It used less material, had a smaller mould surface and did not need as much pre-stressing. At this place, the parameters that could not be taken into account by the algorithm are considered as well:

- Connections.
- Shape of the mould.
- Manufacturing considerations.

The fine structure contains many relatively small members that need to be connected using many connections. The shape of the members in the coarse model seems easier to mould. Because of the smaller number of different elements, the result on the coarse model is easier to build.

The result on the coarse model as depicted in Figure 7.7 performs better than the result on the fine model on every aspect and will be used as a starting point for manually developed design of the supporting elements for the Omnisport hall roof.

7.3 Use of the output

7.3.1 Interpretation

The topology proposed by the algorithm is interpreted as a truss consisting of twelve elements. In the program output the structure is a bit thicker at the node locations. At these locations, peak stresses develop because of members coming together. In the development of the design, a connection strategy has to be developed that enables the efficient transfer of forces from one member to another.

The algorithm output shows a curved upper chord. This might be due to the application of the load to the upper chord. By taking on a curved shape, the force will flow through the elements and bending in the elements will be diminished. The resulting structure can now be interpreted as a truss with a curved upper chord or as a truss with all straight members. These interpretations are shown in Figure 7.9 and Figure 7.10, respectively. Figure 7.11 shows the normal forces in the members for the truss with straight members in the ULS, based on calculations in the finite element program Matrix-Frame. It is chosen to base further calculations on the UHPC truss assuming straight members. One option, where a supported concrete slab is considered, assumes a curved slab. In a later stage of the design the choice for straight members can be reconsidered.

Extra diagonals

The structure as designed by the algorithm is not stable in the middle of the span, where there is no triangulation. Also the lower chord spans 59 metres without being supported. For these two reasons extra diagonals are proposed as shown in Figure 7.9 and Figure 7.10.

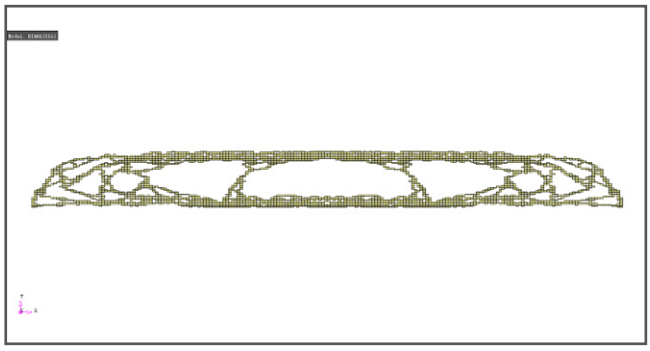


Figure 7.8: Best performing structure emerging from the fine mesh. Picture is mirrored along the line of symmetry.

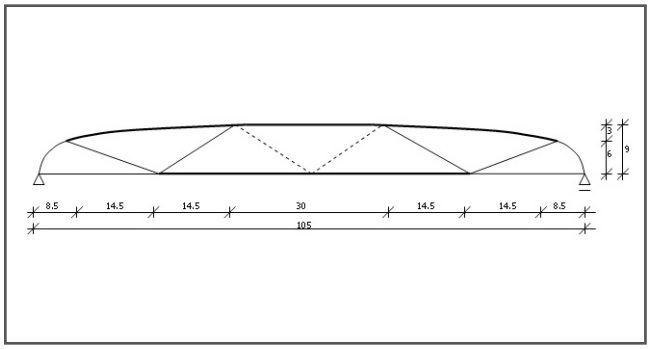


Figure 7.9: Interpretation as a truss with curved upper chord.

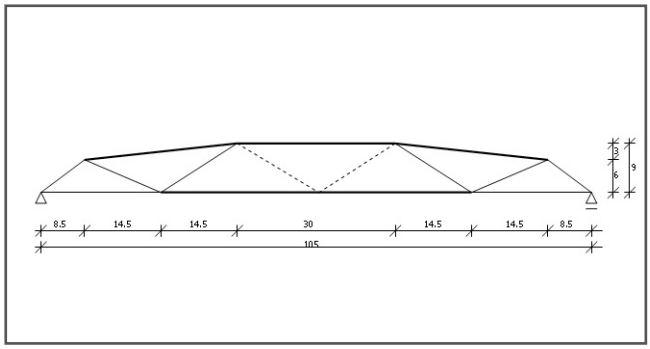


Figure 7.10: Interpretation as a truss with straight upper chord.

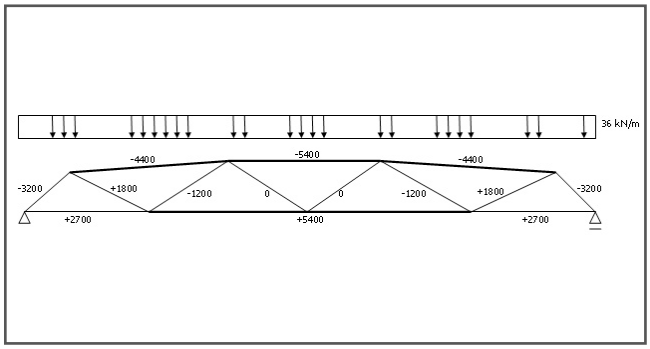


Figure 7.11: Member forces in ULS.

Pre-stress

Figure 7.11 shows that considerable tensile forces occur in five of the members of the truss in the ULS. These elements need to be pre-stressed. The amount of pre-stressing depends on the magnitude of the occurring tensile forces and is depicted in Figure 7.12.

Buckling

Members that are loaded in compression might buckle. For the proposed truss lay-out the sensitivity of the members to buckling is explained in Figure 7.13. The members that are finely dotted are loaded in tension so they are not sensitive to buckling. The upper chord members, the dashed lines, are loaded in compression but are restrained sideways by the roof plate. However, they can buckle in vertical direction in the plane of the drawing. The two diagonals that are loaded in compression are represented with the solid lines; they are sensitive to buckling both in-plane and out-of-plane.

The risk of buckling will have to be taken into account when assigning dimensions to the different members. The members that are sensitive to buckling in one direction can have an I-shaped or a rectangular hollow cross section, whereas the members that can buckle in two directions are better off with a square or circular hollow section.

7.3.2 Options within the result

The ant system based structural design tool has come up with a topology that is interpreted by the user. During this process ideas on how a roof design would actually look like have developed. At this place the different options for developing the roof design are listed and quickly assessed.

The UHPC truss as proposed by the algorithm is taken as a basis for the design. Within this option, questions of how to connect the members arise. Three options are distilled; the pre-stressing can be used to push the members together, the nodes can be cast in concrete thick enough to cope with the peak stresses occurring at these locations, or a concept for a steel node can be developed.

One could also wonder why the ultra-high performance concrete would be pre-stressed, and consider the possibilities for passive reinforcement. As a last option an ultra-high performance concrete roof slab supported by a steel structure that works mostly in tension could be considered.

An overview of this range of possibilities is shown in Figure 7.14.

Steel supported concrete slab

In the analysis of the topology suggested by the algorithm it was shown that five members are loaded in tension. Two diagonals are sensitive to buckling in two directions. The curved upper chord is loaded in compression. This topology could be constructed as a curved UHPC slab supported by a steel structure. Its side view and cross section would then look like shown in Figure 7.15. The steel structure could consist of cables only except for the two diagonals loaded in compression, and the two diagonals in the middle that prevent the system from becoming unstable. They could be made from circular hollow sections to resist buckling. The concrete slab would be a structural element and a roof cover at the same time. The thin slab would be loaded in compression and should be designed such that is could resist buckling.

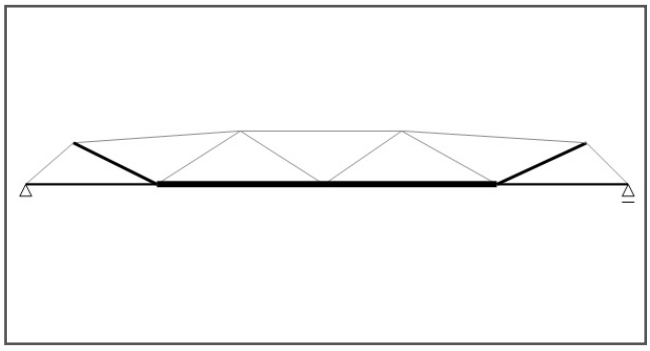


Figure 7.12: Elements that need to be pre-stressed.

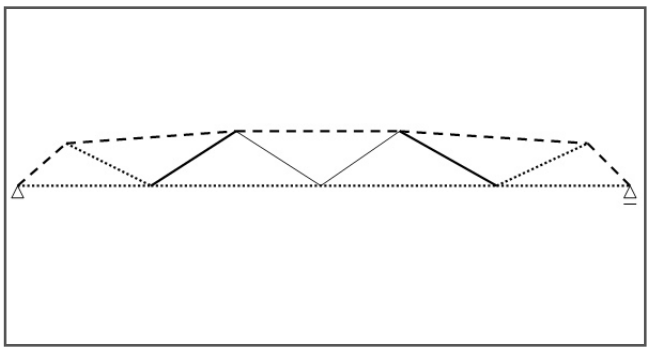


Figure 7.13: Sensitivity to buckling.

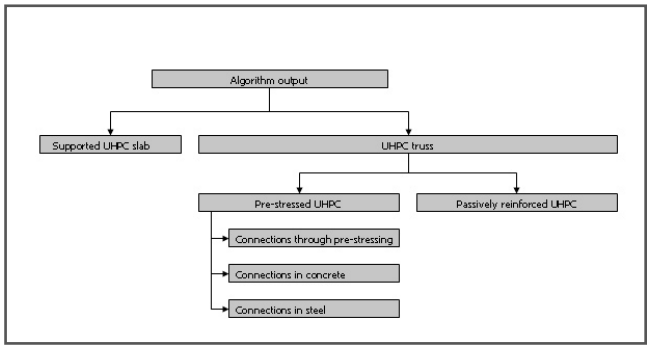


Figure 7.14: Possible options based on the output of the algorithm.

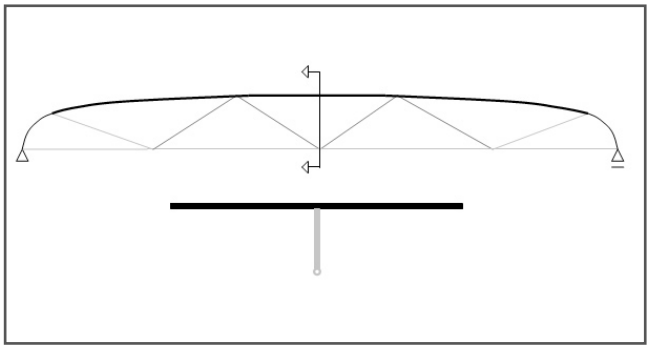


Figure 7.15: Option with UHPC slab supported by steel structure.

Reinforced UHPC

Some may wonder why UHPC is almost always pre-stressed and not passively reinforced. Calculations in Appendix B.3 show that the allowable tensile stress is about twice as large and the modulus of elasticity about four times as large for UHPC when pre-stressed rather than passive reinforcement is applied. These advantages not even take into account the reduction in durability. Passive reinforcement would mean the surface would crack and the concrete become less durable, depending of the amount and distribution of the reinforcement. Because of better performance in terms of capacity, stiffness and durability, the UHPC in this design is pre-stressed.

Pre-stressed UHPC with connections through pre-stressing

Pressing elements together through continuous pre-stressing is a technique used for modular, pre-stressed structures [32, 48, 66]. Figure 7.16 shows which pre-stressing strands should be continuous in order to link the pre-stressed members that are used in the design in UHPC. Two problems emerge if this connection strategy is chosen.

Firstly, some of the pre-stressing strands need to be kinked at the point where the elements are connected. These kinks will cause point loads acting on the structure. These point loads are shown as arrows in Figure 7.16. The extra load on the structure was not taken into account during the optimisation process and cannot easily be coped with in the structure.

Secondly, this connection strategy only links five members. The connection with the other members is not yet secured.

On a whole this connection strategy introduces new issues without providing a connection strategy for the whole structure. Therefore this connection strategy will not be used in the design.

Pre-stressed UHPC with connections in concrete

To make a structure that resembles the algorithm output best, the structure should be made completely in UHPC. Three problems now arise: the manufacturability, the anchoring of the pre-stresses and the force flow in the nodes.

The concrete element is 105 metres wide and 10 metres high. Casting the whole structure in one go seems irrational. Prefabrication of such an element would not be possible because the element would not be transportable. Logistical problems could occur if the concrete would be cast on site. Besides this, shrinkage of the concrete could cause deformations and stresses and post-tensioning of the lower chord of the truss would cause a bending moment on the truss and thus tensile stresses in the upper chord of the truss.

If the structure is cast in one go, the application of a bigger pre-stressing force in one part of the element than in another part is difficult to accomplish.

These problems could be avoided when members are separately casted and assembled through connections that transfer the loads using compression only. Such a connection strategy is depicted in Figure 7.17. This strategy could work for the compressive connections. However, if an occasional upward loading occurs this strategy does not give a satisfying solution and at those places where tensile connections exist, the strategy does not work and difficult connections occur.

Pre-stressed UHPC with steel connections

A last option is to connect separate, pre-cast and pre-stressed concrete elements by steel connections. Such a

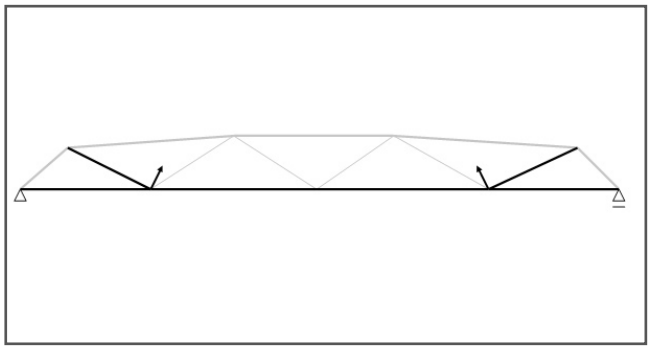


Figure 7.16: Option with UHPC members linked through pre-stressing.

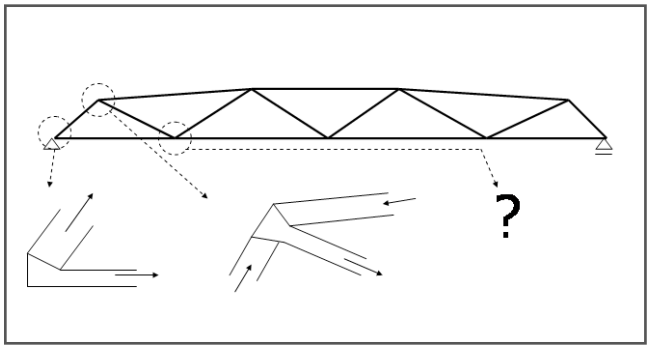


Figure 7.17: Option with UHPC members and concrete connections.

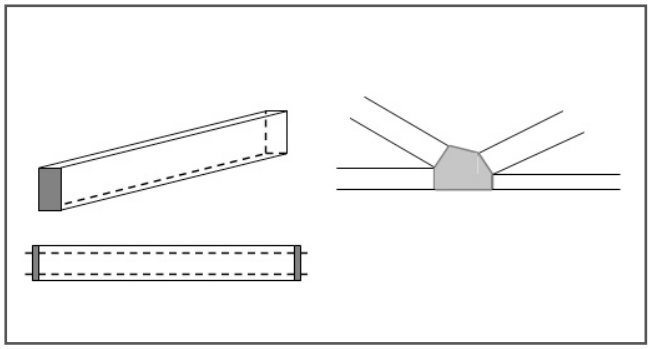


Figure 7.18: Option with UHPC members and steel connections.

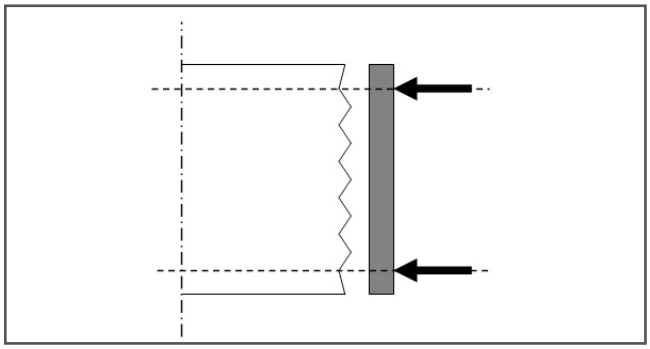


Figure 7.19: Close up of the concrete member and the steel plate (1).

connection would look like the sketch in Figure 7.18. Every member is cast separately. The post-tensioning (dashed line) is anchored on steel plates (depicted in dark grey) that are placed at both ends of the member. This should be possible for this type of concrete, as will be explained later. Now when all the members have a steel plate at their ends, these end plates can be welded onto a plate that is depicted in light grey in the picture and oriented perpendicular to the end-of-member plates. This plate onto which the end plates are connected forms the connection between the members.

If this strategy is adopted, all members can be pre-stressed to a level that suits their situation. Pre-stressing of one member does not influence pre-stressing of another member. Members that do not need a pre-stressing due to the force flow in the structure are pre-stressed to a very low level to connect the plate to the member. This low pre-stressing force will help the member cope with the stresses that might occur during assembly.

As a last remark on this proposed connection strategy, the mechanical behaviour on micro-level in between the plate and the member is explained. Normally an anchor would be cast into the concrete element. Casting concrete around the anchor provides a good contact area. The anchor itself acts like reinforcement to take up the stresses due to the point load that is applied to the structure.

The steel plate that is proposed in this option is assumed to be relatively stiff. The point loads applied to the plate will be transferred to the concrete as a distributed load over the contact area. Because the plate is not cast with the concrete, the contact surface will not be perfect. The microstructure of the concrete area is rough as depicted in Figure 7.19. The steel plate is pushed onto the concrete surface. The grains at the surface of the concrete will be crushed due to the applied stress. Because the ultra-high performance concrete used in this case has a ductile behaviour, the concrete will be able to cope with this small-scale crushing and yield locally in area that is depicted as grey in the figure. A distributed load on the concrete member will result. Figure 7.20 shows the resulting stress on the member and the plate.

7.3.3 Choice of one option

Only two feasible options remain of the options given earlier: the concrete slab supported by a steel structure and the concrete members connected by steel plates.

The idea that the top member of the truss that results from the algorithm can be constructed as the roof cover is interesting. The resulting structure, however, consists mainly of steel. Only the roof cover will be made in UHPC. This thesis is aimed at designing an ultra-high performance concrete structure that supports the roof of the Omnisport roof hall. Taking this into account, choosing this design and refining it does not make sense.

The concept of UHPC members connected by steel nodes makes a truss-like UHPC structure possible. Such a structure fits well into the scope of the thesis. This option is therefore chosen for refinement in a preliminary design.

7.4 Preliminary design

The preliminary design that is elaborated in this paragraph is based on the option that is chosen above. UHPC members are linked by steel connections. The steel connections have conceptually been discussed in the previous paragraph. For this thesis it is assumed that such a concept works. This will have to be checked if this design will actually be applied in practice. The refinement of the concept into a preliminary design as discussed here consists of finding dimensions for the members, giving an estimate of the amount of pre-stress needed, a strategy for its application and ideas about the fabrication.

7.4.1 Load checks

Loads as an input for the algorithm have been calculated in Paragraph 7.1.2 and are checked at this point. The volume of the structure generated by the program is 26.8 m^3, which yields a self weight of 750.4 kN for the complete span. It is assumed that 10% extra material is needed for extra members like the diagonals in the middle. Another 15% of extra weight is assumed for the connections. This yields a total weight of 938 kN for the complete truss, equal to 8.9 kN/m or 0.74 kN/m^2 if the self weight of the truss is assumed distributed over the roof area.

Loads are now:

- Self weight: 0.74 kN/m^2.
- Applied load: 0.80 kN/m^2.
- Snow load: 0.56 kN/m^2.
- Wind load: -0.95 kN/m^2.

This results in the following load cases.

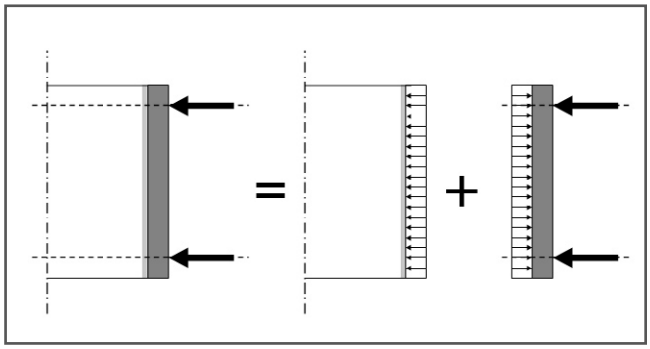


Figure 7.20: Close up of the concrete member and the steel plate (2).

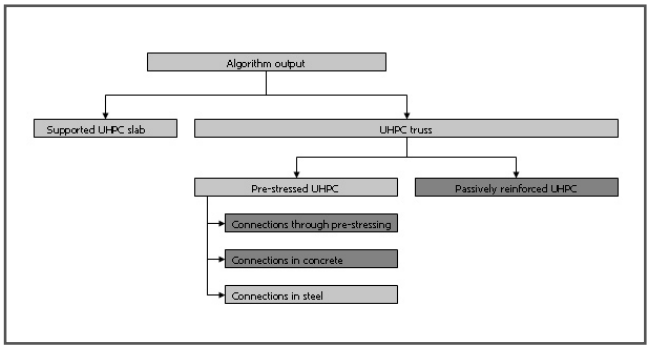


Figure 7.21: Only two of the possible options are feasible.

- SLS snow: *0.74+0.80+0.56 =* 2.10 kN/m^2
- ULS snow: *1.2*(0.74+0.80)+1.5*0.56 =* 2.69 kN/m^2
- SLS wind: *0.74+0.80-0.95 =* 0.59 kN/m^2
- ULS wind: *0.9*(0.74+0.80)-1.5*0.95 =* -0.04 kN/m^2

Calculations in MatrixFrame have been performed based on the ultimate limit state with snow. Figure 7.22 shows the biggest occurring axial forces that will be used for designing the members.

7.4.2 Dimensions

The dimensions of the structure are determined according to the axial forces that are found with the MatrixFrame calculations shown in Figure 7.22. In pre-stressed concrete design, the dimensions are sometimes made based on the loading in the serviceability limit state. The behaviour in the ultimate limit state is then checked, so that the ductile behaviour of the steel can be used for the taking up of the extra loading in this case [17]. Because of the different behaviour of UHPC in comparison with ordinary concrete, especially in terms of ductile behaviour of the concrete, it is not very clear whether this would be a good approach for the case considered in this thesis. It is decided to estimate the dimensions that are needed for the members in this design based on the loading in the ultimate limit state with snow. The mostly centrically pre-stressed elements will need to be able to cope with the loadings within their linear elastic range. This might lead to a conservative estimation of the profiles of the members needed for the design. Refining the size of the members might be possible, but requires further studies on the behaviour of centrically pre-stressed ultra-high performance concrete elements and is not a part of this thesis. It is assumed that the load case with wind suction does not cause any problems, because it will result in a very small upward load or a very small downward load.

Of the four load cases that are determined, one yields suction on the roof due to wind. This load, 0.04 kN/m^2, is neglected in the calculations done in this paragraph. It is assumed that such small stresses can be coped with by the structure because of the tensile capacity of the material.

Pre-stress

For the pre-stressing of concrete steel strands are used that are either pre- or post-tensioned. Pre-tensioned strands need a certain anchorage length to transfer their stress onto the concrete [17]. In the case of the chosen design, the members have to be able to cope with tensile stresses over their full length which means pre-tensioning is not possible.

The rough approximation of the allowable pre-stressing that was done when setting up the algorithm for the Omnisport design problem is refined at this stage. Calculations are done based on the applicable Dutch code [6] and can be found in Appendix B.4. The calculations show that the maximum initial pre-stress is 81 MPa. Losses is pre-stress will reduce this number over time with 17% to 67 MPa. This value is considerably lower than the value assumed when setting up the algorithm. Losses in the pre-stressing steel will lower the effective stress in the strands to *(1-0.17)*1350 =* 1120 MPa.

Member dimensions

Members are assigned the letters A until H. Each member will be shortly discussed; a cross section and pre-stressing are determined. The determination of the member dimensions are based on the forces calculated in MatrixFrame and shown in Figure 7.22 and the buckling lengths shown in

Member	Width (mm)	Height (mm)	Notes
A	500	200	Rectangular hollow section, tapered over the length
B	750	250	Rectangular hollow section
C	750	250	Rectangular hollow section
D	125	250	Rectangular solid section
E	330	330	Square hollow section
F	150	150	Square solid section
G	120	400	Rectangular solid section
H	175	500	Rectangular solid section

Table 7.1: Proposed dimensions.

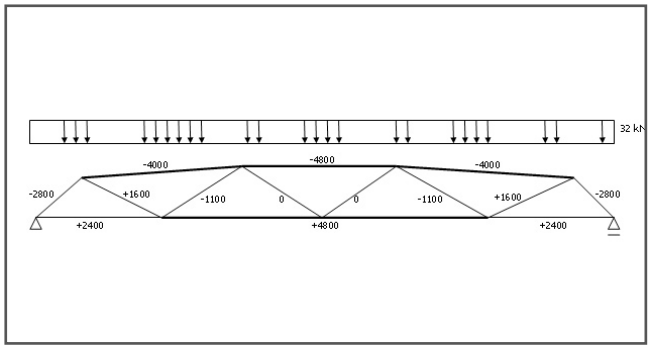


Figure 7.22: Axial forces resulting from the ultimate limit state with snow.

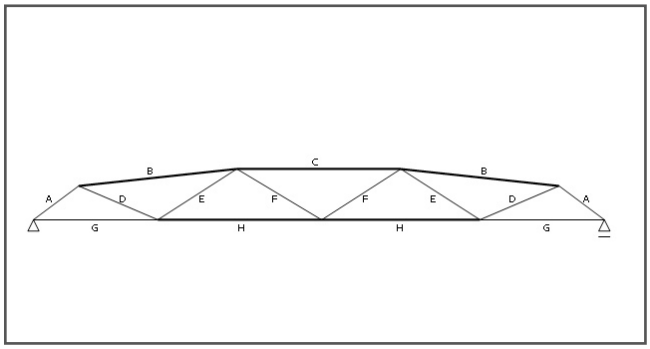


Figure 7.23: The members are assigned letters.

Figure 7.24. At this place, members are calculated as either rectangular hollow or solid sections because of the author's preference. It is explicitly noted that other cross sectional shapes like I-beams can also be applied; calculations in this paragraph are indicative.

Proposed members and the calculation of their cross sectional properties are given in Appendix E.2. Resulting dimensions are given in Table 7.1.

Dead load compensation

The members are calculated based on pure axial stress. Dead load consisting of self weight and, in case of the upper chord, applied load, will have to be accounted for. This can be done by curving the pre-stress such that an upward distributed load is generated with the same magnitude as the dead load. It is proposed to apply this strategy to the lower chord of the UHPC truss that is loaded in tension. A quick calculation is done for member H. This member has a cross sectional area of 875,000 mm^2 and thus a self weight of 2.45 kN/m. This results in a bending moment in the middle of the span of $1/8*q*l^2$ = 276 kNm. The applied pre-stress is 4800 kN and needs to be positioned in the beam with an eccentricity of 57 mm to compensate the bending moment due to the dead load. This principle is sketched in Figure 7.25.

Further calculations

The members forming the upper chord of the UHPC truss are not only loaded by their self weight but also directly carry the roof plates and their superimposed loads. These members might need a bigger depth, cross sectional area or pre-stress in order to be able to cope with the bending following from these loads. This further detailing is left to a next stage of the design and is not considered in this thesis.

Further calculations and checks will be necessary to confirm or refine any of the members. In those calculations, stresses and displacements due to bending, self weight and the other load cases like wind are necessary.

Special attention will need to be given to durability issues and fire behaviour. The durability of the ultra-high performance concrete has not been checked as a part of this thesis. For the time being it is assumed that durability issues are not a problem for this material, and that pre-stressing strands will have enough protection even when a relatively small coverage is used. This assumption is based on the findings in Paragraph 4.4.2. Further research is necessary to be able to validate this assumption. If the research shows that any durability issues for the concrete of the pre-stressing steel prove problematic, coatings like those used in the steel industy might provide a solution.

Similarly, fire resistance has also not been researched as a part of this thesis. Further research is necessary. If this issue proves to form a problem for the design, again coatings like those used in the steel industry might provide a solution.

7.4.3 Economical feasibility

It is interesting to estimate the costs of the truss that has been designed. Because the member sizes are know, the amount of concrete that is needed for the structure can be calculated. Results are given in Table 7.2.

The calculated value of 21.4 m^3 is less than the amount given by the algorithm; 26.8 m^3, but the calculated value can change slightly when checking member sizes and excludes the weight of pre-stressing steel within the members. Besides this, extra weight is added by connections. Material costs for 21.4 m^3 of UHPC are around € 43,000. It is assumed that the costs for pre-stress, nodes, mould, fabrication and placing of the elements are of around the same amount. This yields a total estimated cost of around € 86,000 for one truss. This is a very rough estimate, but can be compared to the costs of the steel truss in the current design.

The steel in the current design weighs around 0.50 kN/m^2 or 63,000 kg for a truss supporting 105 by 12 metres. Assuming one kilogram of steel has an in-place cost of around € 2, the costs of one steel truss are estimated on € 126,000.

It is explicitly noted that the UHPC design is rough and the estimated costs are not more than an educated guess.

Member	Area (m^2)	Length (m)	Volume (m^3)
A	0.050	2 x 10.4	1.04
B	0.094	2 x 29.2	5.49
C	0.120	30.0	3.60
D	0.031	2 x 15.7	0.97
E	0.073	2 x 17.1	2.50
F	0.022	2 x 17.5	0.77
G	0.044	2 x 23.0	2.02
H	0.085	2 x 29.5	5.02
Total			21.4

Table 7.2: Volume of concrete in the truss.

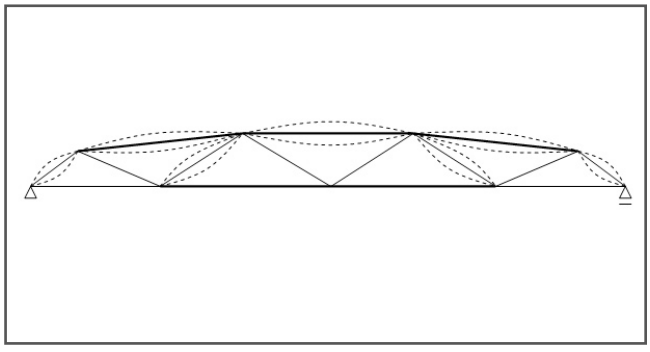


Figure 7.24: Buckling lengths of the members.

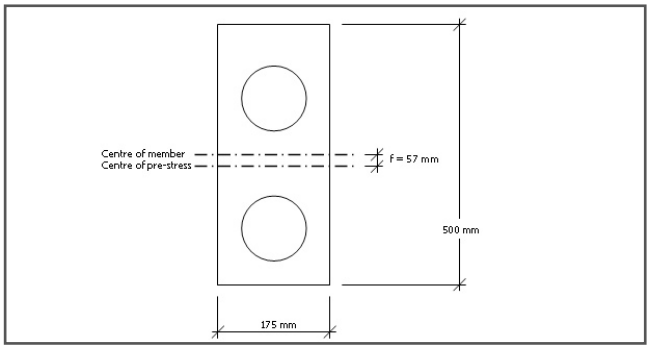


Figure 7.25: Dead load compensation by eccentrically placed pre-stress.

However, the estimated costs of the steel and concrete structures are in the same order or magnitude. The estimation is too rough to claim that this proves that a UHPC structure is economically feasible when compared to a steel structure, but it seems unlikely that a UHPC structure will cost many times more than a the option in steel.

7.4.4 Resulting structure

Stiffness

When the section sizes as calculated are implemented in a MatrixFrame analysis, the biggest displacement in the serviceability limit state due to self weight, dead load and snow load occurs in the middle of the span and is 0.24 metres. This is equal to span/446 and is assumed acceptable.

Connections

The steel connections as described in Paragraph 7.3.2 will be applied in this preliminary design. The concept will need to be developed and checked before it can be introduced in the actual structure. For this preliminary design it is assumed that a connection strategy like this is possible.

Fabrication

The members can be cast in a factory. Hollow sections can be fabricated in the same manner as the roof of the Millau viaduct toll gate has been fabricated; by casting the concrete around polystyrene foam blocks. In order to meet the assumptions done on pre-stress losses, the concrete has to harden for 28 days before being pre-stressed, transported to the site and lifted into place.

Other trusses

Only one of the trusses that support the roof surface is redesigned. This truss is governing because is has the largest span. The other trusses are not considered in this thesis. Proving the structural feasibility of the governing truss when redesigned in UHPC implies that the other trusses can be redesigned in UHPC as well. The actual design of these other trusses is left to a next stage in the design process and is not part of this thesis.

Influence on the roof shape

The proposed truss layout leads to a different shape of the Omnisport hall roof surface. In the original design the roof surface was curved in one direction. The steel trusses support the roof in its plain direction. Because of the curvature of the upper chord of the truss, the roof surface now curves in this direction as well. This either results in a roof surface with double curvature or in the same roof surface, but then curved in the other direction.

7.5 Conclusions

A load-bearing structure for the Omnisport hall roof in ultra-high performance concrete is possible. Based on the findings of the structural topology design tool that has been developed earlier in this thesis, two types of structures that span the largest cross section of the roof proved possible.

Firstly, a concrete slab supported by a steel structure can be applied. The concrete slab covers the roof and also forms the upper chord of a truss. The slab is loaded in compression.

A second well-performing option is a truss-like structure in UHPC. Of the several fabrication and connection strategies that have been researched, the option with steel connections is the most promising. Further research and development has focused on this option.

Elements are constructed separately in a factory. After casting and hardening, steel end plates are attached to the elements and function as an anchoring for the pre-stressing strands. The end plates are welded together. The pre-stressing forces in one element do not cause loads on other elements in this manner.

Calculations that are performed in this chapter are rough and need to be checked and refined in a later design stage that is not part of this thesis. Calculations are only performed on the governing span in the design. The other trusses in the design are to be developed. The shape of the trusses causes a change in the shape of the roof of the Omnisport hall.

A preliminary estimation of the costs of the truss that is developed shows that an option in ultra-high performance concrete is not necessarily more expensive than an option in steel.

CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS

This chapter elaborates on the conclusions and recommendations that are drawn based on the work in this thesis.

8.1 Conclusions

On the algorithm

- A new application for the ant colony optimisation metaheuristic has been developed for multi-objective structural topology optimisation. The algorithm is capable of performing a trade-off between different performance parameters that are contradictory.
- The algorithm is successfully validated with least weight problems. Visual comparison shows that the obtained topologies are similar to a priori expected Michell trusses. The algorithm is applicable to structural design cases.
- The developed algorithm is robust for the different problems and boundary conditions it has been applied to. Meshes with up to 5250 elements have been tested and several different problems have been considered. On each problem, sound results are found.
- The algorithm is capable of finding an optimal structural topology for the test case as considered in this thesis.
- The risk of premature stagnation can be reduced at the cost of slower convergence. Run times go up with the number of elements analysed. Because of this trade-off, the tool has proven especially valuable for coarsely meshed two dimensional problems. The process is then quicker and less sensitive for random fluctuations causing small-scale discrepancies.
- The critical mindset of an engineer is necessary when using the output of the algorithm. Due to the limitations in the representation of the problem that is optimised for, the output might not be applicable in practice. Examples are buckling, stability considerations and special load cases.

On the design case

- A load-bearing structure for the Omnisport roof in ultra-high performance concrete is possible and is not necessarily more expensive than an option in steel.

On ultra-high performance concrete

- For ultra-high performance concrete, structural systems that are different from those used for ordinary concrete become efficient. This is because of the superior properties of UHPC. An example of a structural system that is efficient for UHPC is the truss.
- Pre-stressing is the best of the studied methods to cope with the tensile stresses in ultra-high performance concrete structures in an efficient way.

8.2 Recommendations

On the algorithm

- The tool should only be used for topology optimisation. Resulting structures are merely indicative and give a very rough visualisation of a possible optimum structure. Elements have to be distilled and the structure should be checked, refined and further developed before it can be implemented in practice.
- Refine and improve the developed algorithm, especially focusing on the following:
  - *Create an interface for the developed algorithm to improve the user-friendliness of the tool.*
  - *Improve the connectivity analysis.*
  - *Extend the structural analysis with a check on stability or load sensitivity.*
  - *Research the possibilities to implement additional techniques that add structural knowledge to the algorithm. Such techniques could include the recognition of structural elements in the generated solutions and techniques that make sure that only feasible structures are tested.*
  - *Programmers and FEM specialists should check and optimise the code of the algorithm in order to diminish run times.*
  - *Extend the algorithm to three-dimensional problems. The concept suits such problems and parts of the algorithm are already able to cope with three dimensions.*
  - *Enhance the random seed in the algorithm. At the moment, the seed is not unique for every structure.*

- Create a different tool for the shape and size optimisation of the structures that result from the developed topology optimisation algorithm. Such a tool could be based on a local search strategy. Re-defining and meshing the design space after finding a topology could also be effective.
- Develop alternative optimisation algorithms for the same purpose, for example using a generic algorithm. This is the only way of enabling one to draw conclusions on the efficiency of the algorithm.

On the design case

- The structure that is designed based on the algorithm's output should be developed further. The current design is preliminary and only focuses on one of the trusses. It is not ready for direct implementation in the rest of the design. Further development consists of several issues:
  - *The derived member sizes should be refined.*
  - *Issues and load cases that have not been considered in this thesis should be checked.*
  - *Similar structures for the other trusses that form part of the load-bearing structure of the roof should be generated.*
  - *The structure should be integrated with the rest of the design; connections to columns and application of the roof cover should be considered.*
- Generate similar structures for the other trusses in the lay-out. The current design is preliminary and cannot be directly implemented as part of the structure. It only focuses on one of the trusses in the design.
- Develop and research the connection strategy with steel connections that is proposed in this thesis. If the strategy proves feasible, it could be a cheap and handy connection strategy for ultra-high performance concrete elements.
- Determine the costs of the preliminary designed ultra-high performance concrete structure for the Omnisport roof more accurately. This is necessary to draw conclusions on the economical feasibility of the developed design.

8.3 Vision

The work in this thesis focuses on recent developments in the field of computational optimisation and materials science. The work has resulted in a vision for the future application of both optimisation techniques for structural design and the future application of ultra-high performance concrete.

On computational optimisation

The algorithm that is developed for this thesis functions, but it needs a very long time to run. This is not necessarily a problem, taking three considerations into account. Firstly, computers are becoming quicker and quicker. Processor technology as well as parallelisation techniques develop. It is probable that this process continues in the next decades. An algorithm as developed in this thesis might need a week to run at the moment, but if the development in processor and parallelisation technology continues it might be that the same optimisation process can be done in an hour in ten or twenty years time. Besides this, it is concluded that the algorithm leaves room for improvements on its efficiency. These improvements might lead to diminishing of run times just by enhancing the algorithm's performance. Lastly, one has to question whether it is really a problem if runs take a long time. The purpose of the algorithm is to inform the engineer or designer about possible optimal topologies for structural design problems. Taking this purpose into account, it is not necessarily a problem that the routine might have to run overnight or over the weekend; it can still suit its purpose.

The explanation above is not intended to suggest that optimisation tools do not have to be quick. In fact, the vision of the author is that two types of digital structural design tools will emerge; very quick parametrical design tools for quick assessment of structures as well as tools that generate possible solutions and suggest those to the user. For the latter group it is not a big problem if the process takes a considerable amount of time.

Just like the introduction of finite element software did not mean that the profession of the engineer could be taken over by computers, tools like the one developed in this thesis do not in any way make the profession of the engineer unnecessary. In fact, a good engineer is essential to interpret the results of such a tool. A tool, at the end of the day, is merely a tool. It is a piece of software that outputs an optimal solution for the analysed problem, but it is not capable of fully assessing the solutions. It might come up with a solution that does not even work for the full set of boundary conditions. It will only take into account the constraints, boundary conditions and load cases that were actually implemented in the run.

Computational optimisation algorithms inform the engineer about possible optimum structures for a specific problem. The engineer has to make the decision whether such a structure can actually be implemented and if it is actually optimal, also with respect to issues that were not part of the analysis. If the structure is chosen, it will need further development and assessment. Information on possible optimal structures can be especially useful when uncommon materials are used or when special boundary conditions or spans are considered.

On ultra-high performance concrete

The material ultra-high performance concrete is relatively new. Its superior performance when compared to ordinary concrete gains attention. Possibly, developments in the field of concrete science will enable an even stronger, stiffer and more durable type of concrete to emerge.

UHPC is not yet widely applied. The work in this thesis showed that applying UHPC to a structure that is now constructed in steel is possible. It can be expected that more and more structures will be constructed in this material, especially now when the steel price is going up and a code for UHPC is supposed to be finished shortly.

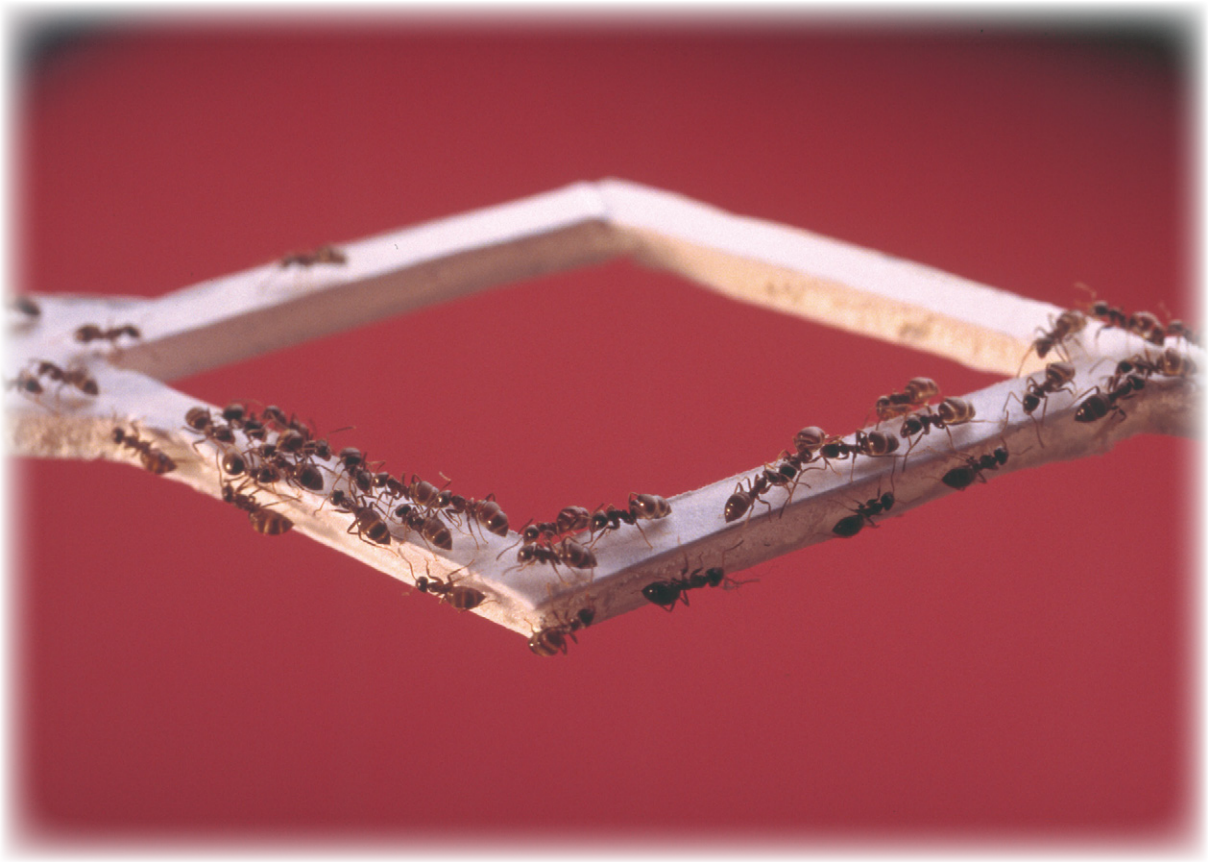
The work in the thesis shows that new structural topologies can evolve when such a new material is assessed. Taking this into account, engineers and architects face the challenge of finding the right structural shapes and right applications for this material.

This thesis does not focus on sustainability considerations, but it is important to keep in mind that this is an important issue. Sustainability issues might affect the use of UHPC, depending on its sustainability indicators, on politics and on

the development of the global warming process. It is impossible to do any claim about the sustainability indicators of UHPC because this has not been researched in this thesis.

ANT SYSTEM BASED STRUCTURAL DESIGN OF A ROOF IN ULTRA-HIGH PERFORMANCE CONCRETE

APPENDICES



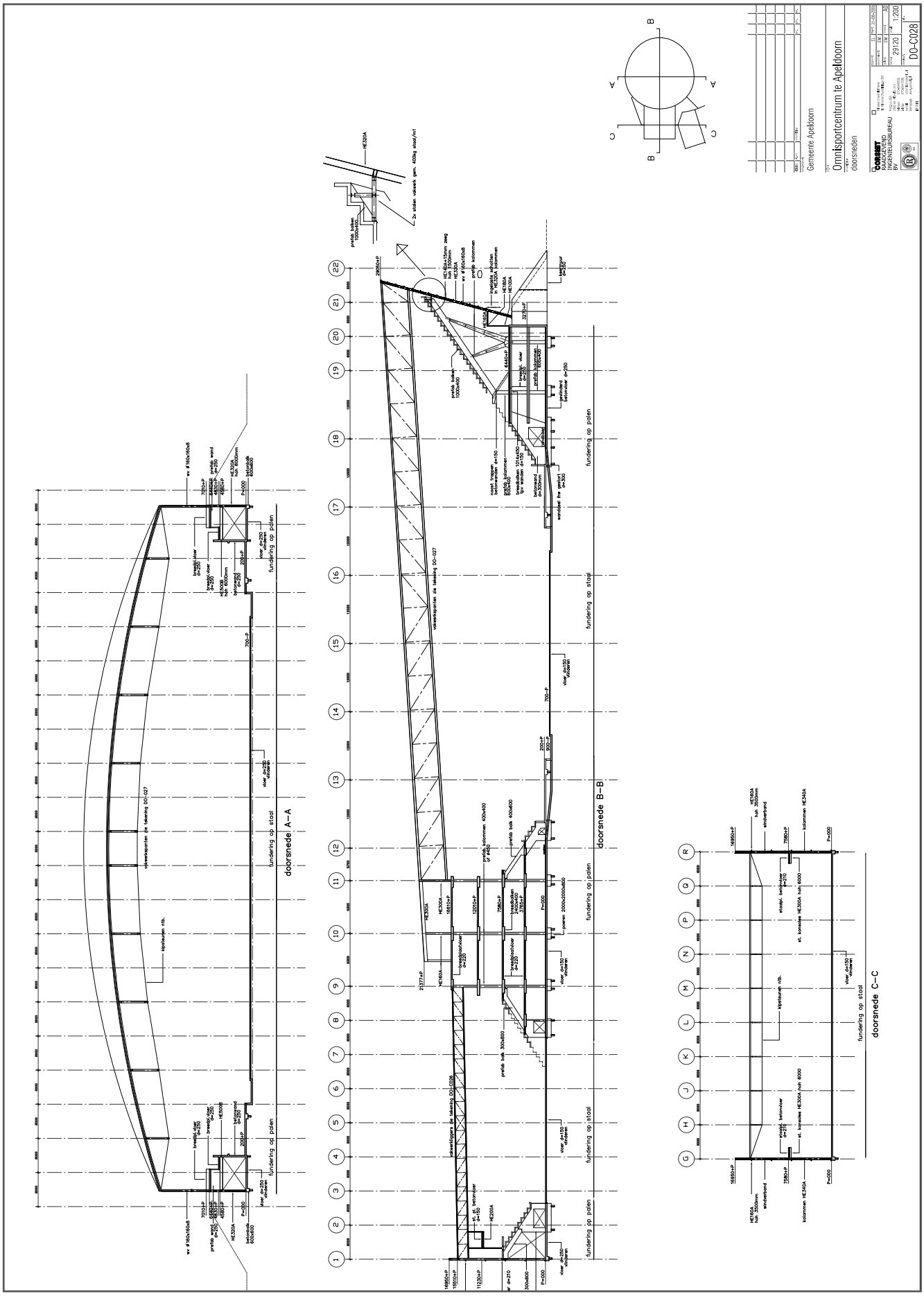
MSc Thesis
Delft University of Technology

MARTIN FLINT

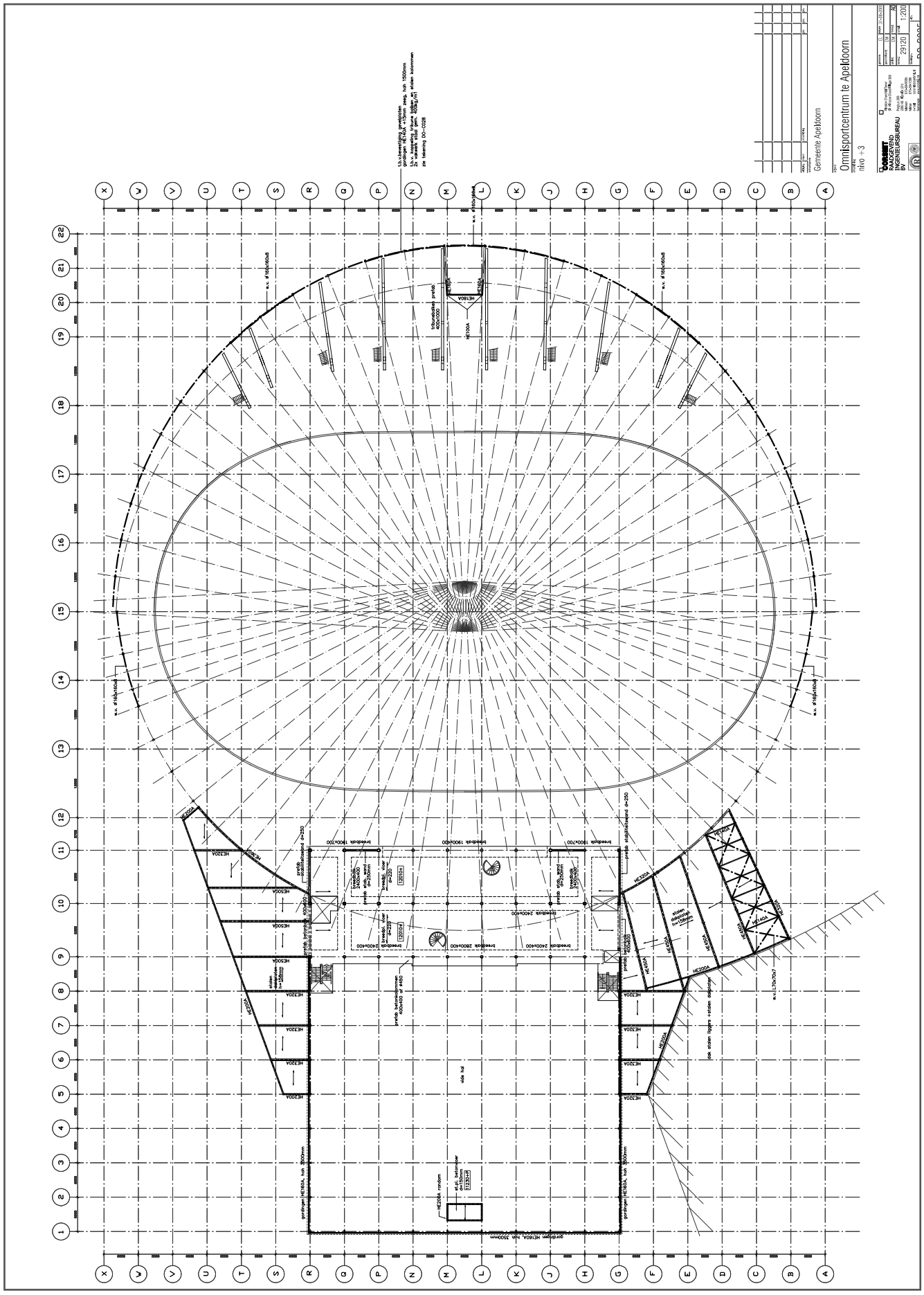
JUNE 2008

APPENDIX A OMNISPORT ROOF

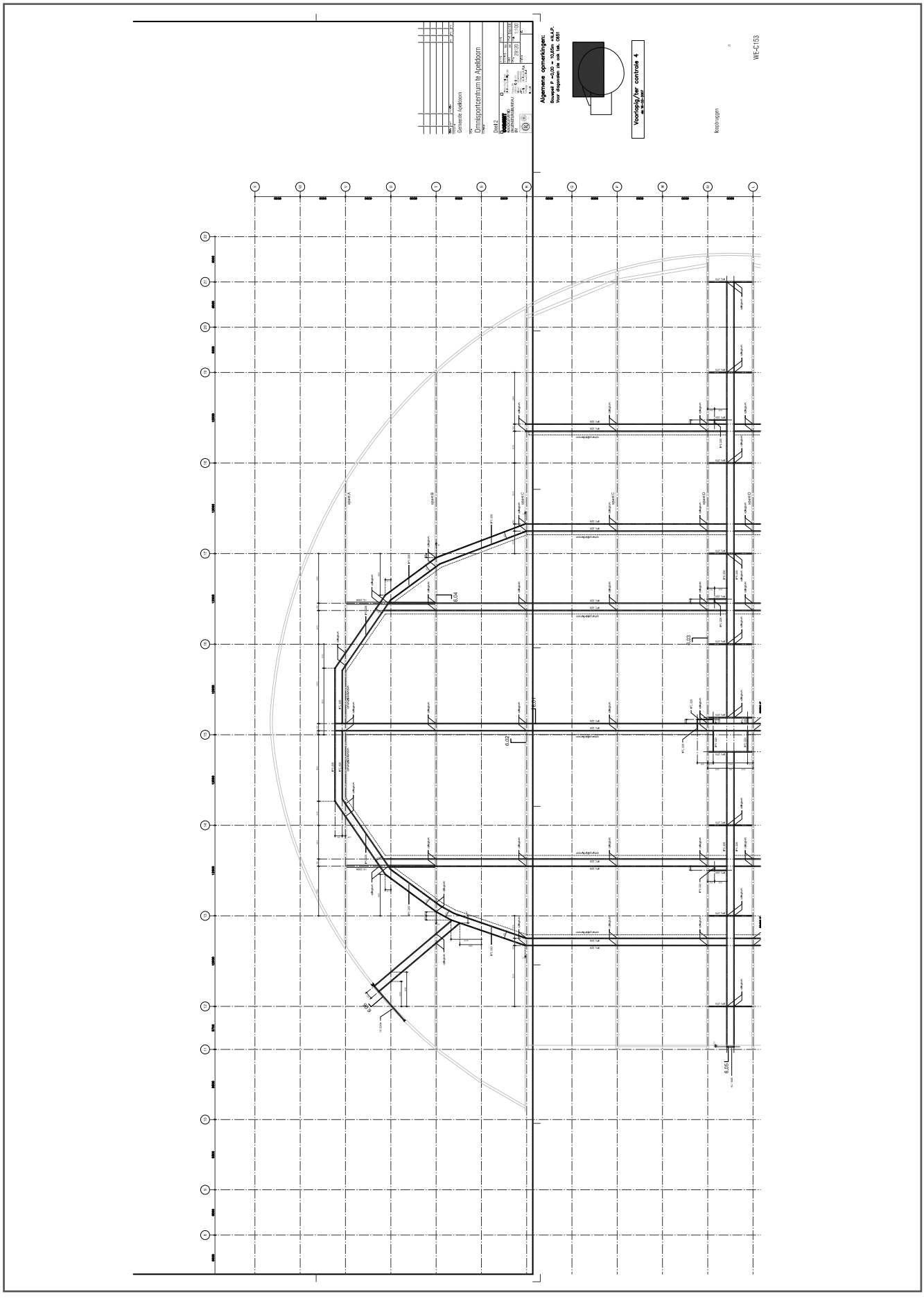
Omnisport hall cross section



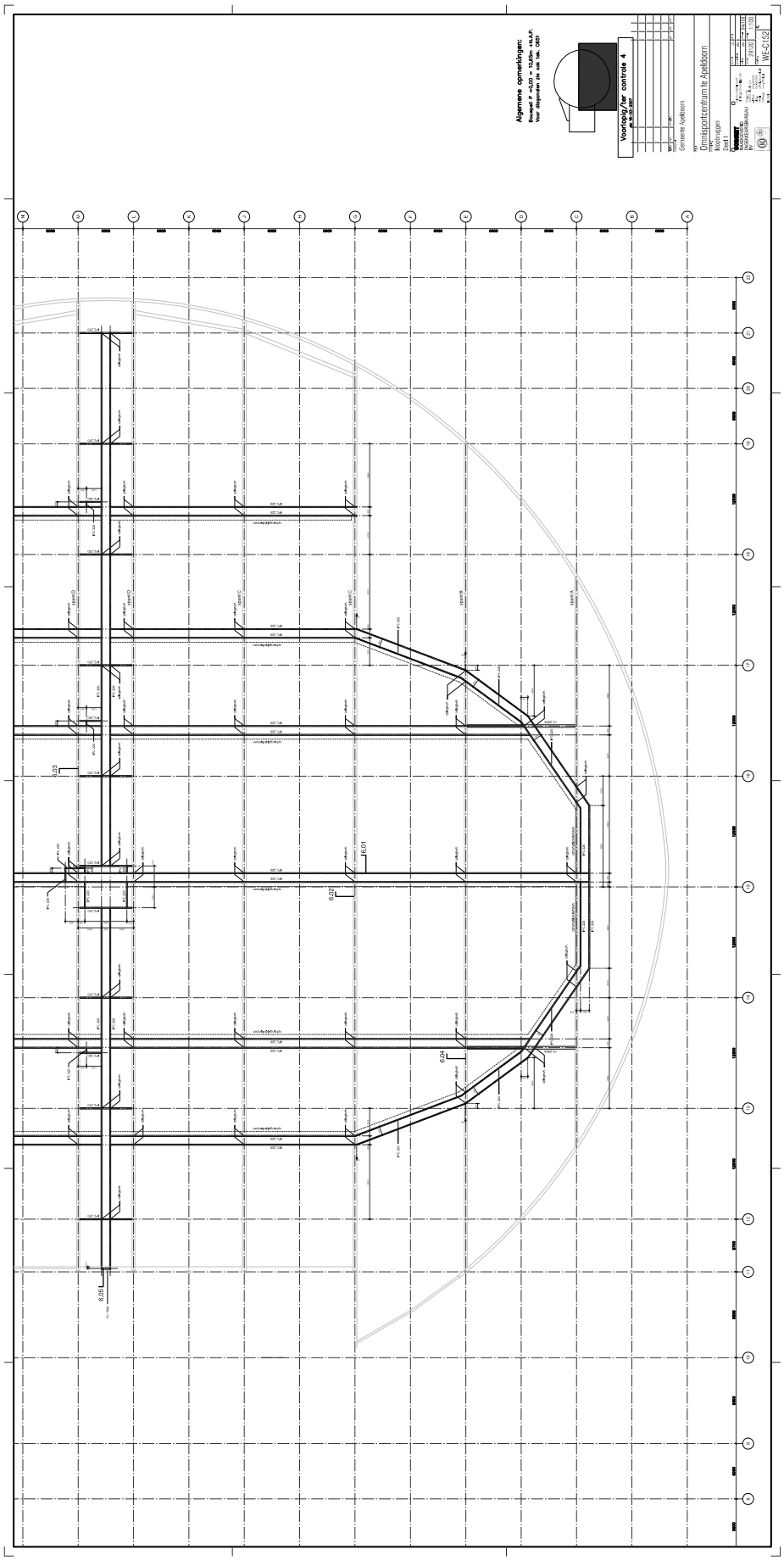
Omnisport hall plan



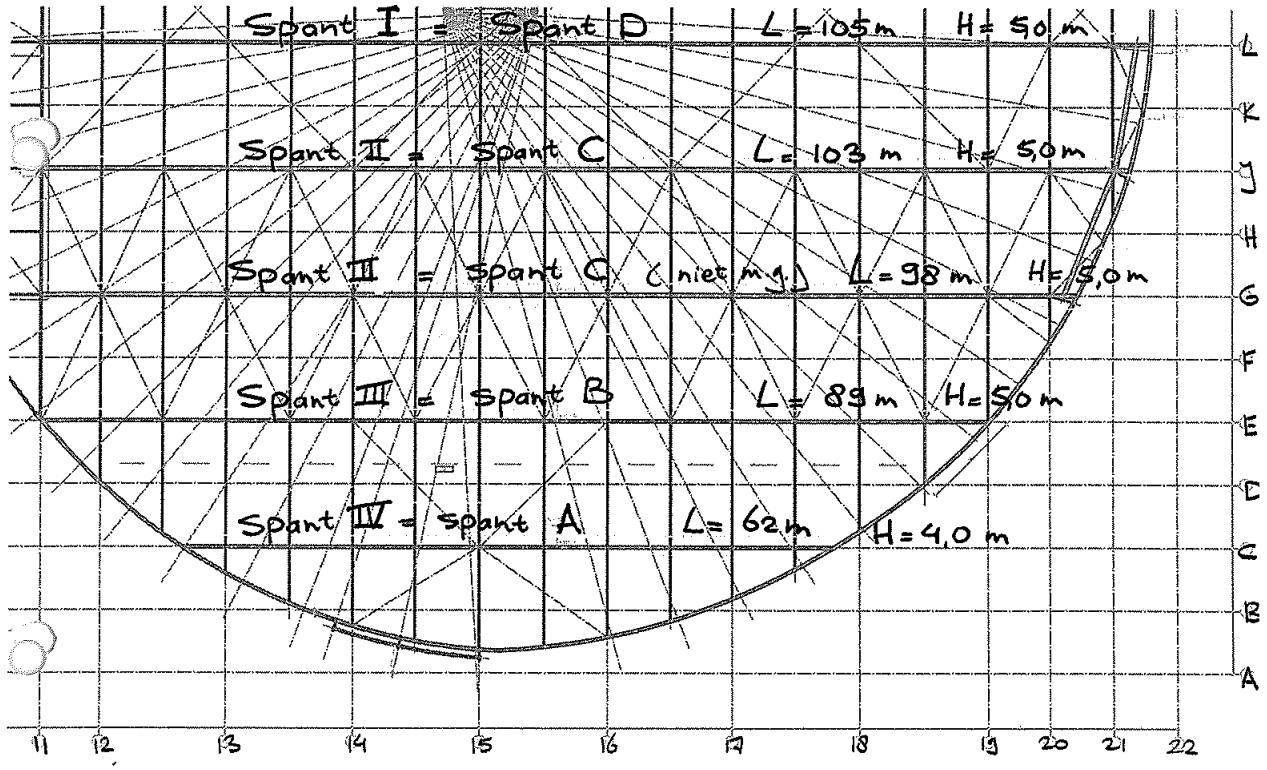
Omnisport hall walking bridges (1)



Omnisport hall walking bridges (2)



Loads determination as done by the engineers



CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. 29120 berek. no. WE-Rg03	blz. no. 150

Omnisport Apeldoorn - dak WAH:

Belastingen: PB	- felsdak (kalzip 3-5 kg/m²)	0,04	kN/m²
	- dakplaten (11 kg/m²)	0,11	"
	- isolatie (5 kg/m² = ruim)	0,05	"
	- 2x 10 mm multiplex (à 500 kg/m³)	0,10	"
	- gordingen + divers	0,20	" (= ruim)
	- Leidingen + verlichting e.d. (externe Lasten aan spant)	0,17	"
	- spanten	0,50	"
	- Loop brug	0,11	"
		1,28	kN/m²
Aangehouden belasting:	p_{PB} =	1,30	kN/m²
VB	- sneeuw	0,56	kN/m²
	- wind (III / b.b. / H = 30 m)	0,95	kN/m²

$C_e = 0,3 + 0,7 = 1,0$

APPENDIX B CALCULATIONS ON CONCRETE

B.1 Stress-strain diagram of UHPC

Characteristic compressive strength	f_{cj} = 180 MPa	(AFGC/Setra 2002)
Modulus of elasticity	E_{ij} = 65 GPa	(AGFC/Setra 2002)
Characteristic direct tensile strength	f_{ck} = 8 MPa	(Hajar 2003)
Char. tensile post-crack strength	σ_{bt} = 9.1 MPa	(Hajar 2003)
Member depth	h = 60 mm	(assumption)
Fibre length	l_f = 15 mm	(conservative assumption)
Characteristic length	$l_c = \frac{2}{3}h$ = 40 mm	(F.R. p 47)
Orientation co-efficient *	K = 1.25	(F.R. p 46)
Material factor **	γ_b = 1.5	(BPEL 91 p 29)
Partial safety factor ***	γ_{bf} = 1.05	(F.R. p 46)
Load duration factor	θ = 1 if $t > 24$ h θ = 0.9 if 1 h $< t \leq 24$ h θ = 0.85 if $t \leq 1$ h	

* The orientation co-efficient takes into account the steel fibre orientation and can be assumed 1.25 for beam-like elements.

** The material factor is obtained from the French code BPEL 91 (revision 1999) for prestressed concrete and is the same as in the EuroCode

*** The partial safety factor is an extra safety factor for a reduction of tensile capacity due to manufacturing defects. For this big-scale and future case, the smallest of the two possible reduction factors is assumed.

F.R. is the abbreviation for the French Recommendations published by AGFC/Setra in 2002

Serviceability limit state

Design compressive strength $\sigma_{bc} = 0.6 f_{cj}$ (F.R. p51)

$= 108$ MPa

According strain $\varepsilon_{bc} = \frac{\sigma_{bc}}{E_{ij}}$

$= 1.66$ ‰

Design direct tensile strength $f_{tj} = \frac{f_{ck}}{\gamma_b}$

$= 5.3$ MPa

According strain $\varepsilon_e = \frac{f_{tj}}{E_{ij}}$ (F.R. p50)

$= 0.08$ ‰

Design post-crack tens. str. $\sigma_{bt} = \frac{\sigma(w_{0.3})}{K \cdot \gamma_b}$ (F.R. p51)

$\sigma(w_{0.3}) = 9.1$ MPa (Hajar 2003)

$\sigma_{bt} = 4.9$ MPa

According strain $\varepsilon_{0.3} = \frac{w_{0.3}}{L_c} + \frac{f_{tj}}{E_{ij}}$ (F.R. p50)

$= 7.6$ ‰

Design 1% crack strength $\sigma_{1\%} = \frac{\sigma(w_{1\%})}{K}$ (F.R. p51)

$\sigma(w_{1\%})$ is unknown and should be measured on a sample

$\sigma_{1\%} = 2.5$ MPa (assumption)

According strain $\varepsilon_{1\%} = \frac{w_{1\%}}{L_c} + \frac{f_{tj}}{E_{ij}}$ (F.R. p51)

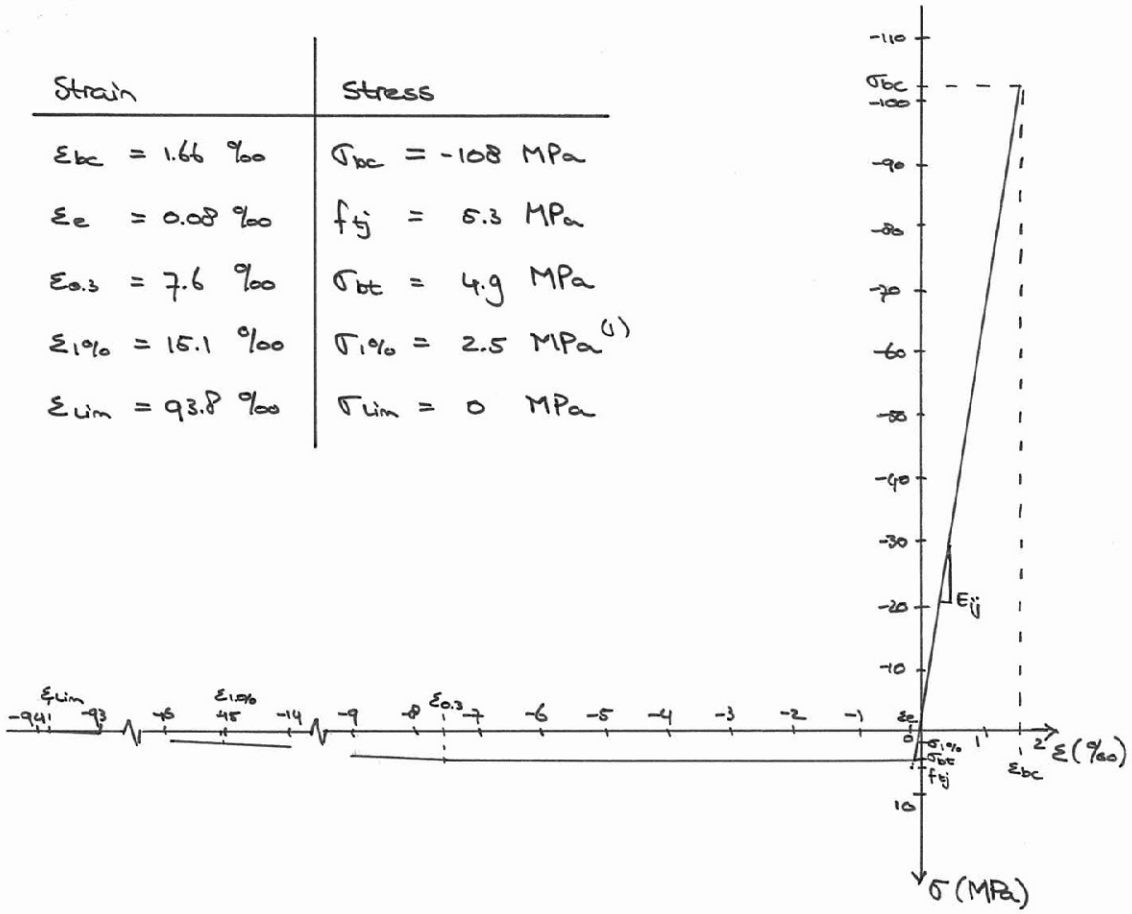
$= \frac{0.01\,h}{L_c} + \frac{f_{tj}}{E_{ij}}$

$= 15.1$ ‰

Ultimate strain $\varepsilon_{lim} = \frac{L_f}{4 L_c}$ (F.R. p51)

$= 93.8$ ‰

Strain	Stress
$\varepsilon_{bc} = 1.66$ ‰	$\sigma_{bc} = -108$ MPa
$\varepsilon_e = 0.08$ ‰	$f_{ij} = 6.3$ MPa
$\varepsilon_{0.3} = 7.6$ ‰	$\sigma_{bt} = 4.9$ MPa
$\varepsilon_{1\%} = 15.1$ ‰	$\sigma_{1\%} = 2.5$ MPa (1)
$\varepsilon_{lim} = 93.8$ ‰	$\sigma_{lim} = 0$ MPa



(1) assumption

Ultimate limit state

Design compressive strength $\sigma_{bcu} = \frac{0.85\, f_{cj}}{\theta \cdot \gamma_b}$ (F.R. p 59)

$= 102$ MPa *

According strain $\varepsilon_{bc} = \frac{\sigma_{bcu}}{E_{ij}}$

$= 1.57$ ‰

Ultimate strain $\varepsilon_u = 3$ ‰ (F.R. p 58)

Design direct tensile strength $f_{tj}/\gamma_{bf} = \frac{f_{tj}}{\gamma_{bf}}$ (F.R. p 58)

$= 5.0$ MPa

According strain $\varepsilon_e = \frac{f_{tj}}{\gamma_{bf} E_{ij}}$

$= 0.078$ ‰

Design post-crack tens. str. $\sigma_{btu} = \frac{\sigma(w_{0.3})}{K \cdot \gamma_{bf} \cdot \gamma_b}$ (F.R. p 59)

$= 4.6$ MPa

According strain $\varepsilon_{u0.3} = \frac{w_{0.3}}{l_c} + \frac{f_{tj}}{\gamma_{bf} \cdot E_{ij}}$ (F.R. p 58)

$= 7.6$ ‰

Design 1% crack strength $\sigma_{u1\%} = \frac{\sigma(w_{1\%})}{K \cdot \gamma_{bf} \cdot \gamma_b}$ (F.R. p 59)

$\sigma_{w1\%}$ is unknown and should be measured on a sample

$\sigma_{u1\%} = 2.4$ MPa (assumption)

According strain $\varepsilon_{u1\%} = \frac{w_{1\%}}{l_c} + \frac{f_{tj}}{E_{ij}}$ (F.R. p 59)

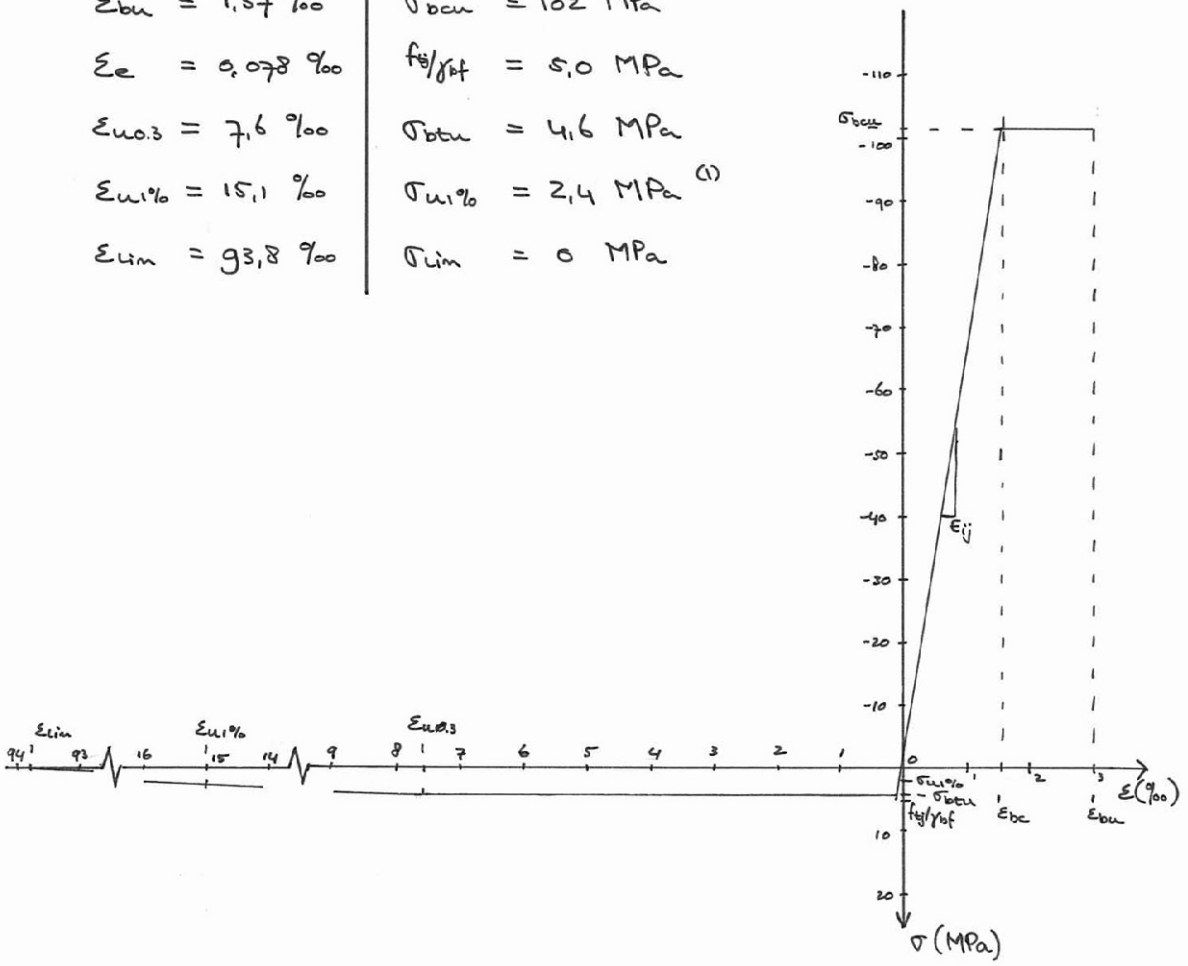
$= 15.1$ ‰

Ultimate strain $\varepsilon_{lim} = \frac{l_f}{4 l_c}$ (F.R. p 59)

$= 93.8$ ‰

* The worst case scenario in terms of loading is assumed : $\theta = 1$
This is inappropriate for e.g. wind loading, where $\theta = 0.85$ and $\sigma_{bcu} = 120$ MPa.

Strain	Stress
$\varepsilon_u = 3$ ‰	$\sigma_{bcu} = 102$ MPa
$\varepsilon_{bu} = 1{,}57$ ‰	$\sigma_{bcu} = 102$ MPa
$\varepsilon_e = 0{,}078$ ‰	$f_{tj}/\gamma_{bf} = 5{,}0$ MPa
$\varepsilon_{u0.3} = 7{,}6$ ‰	$\sigma_{btu} = 4{,}6$ MPa
$\varepsilon_{u1\%} = 15{,}1$ ‰	$\sigma_{u1\%} = 2{,}4$ MPa (1)
$\varepsilon_{lim} = 93{,}8$ ‰	$\sigma_{lim} = 0$ MPa



(1) assumption

Stress values are governing in the Ultimate Limit State

Buckling

The upper members in the structure can be supported sideways by the roof slab.
In between supports this member can buckle
Diagonals and verticals can buckle.

Assume the risk of buckling reduces the allowable compressive stress to $0{,}75\,\sigma_{buc}$.

Prestress reduction

Depends on creep and shringage, which will be less than 20 %, because creep and shrinkage will be less than in ordinary concrete. (Walraven 1997, AFGC/Setra 2002, Servant 2006)

Conservative assumption: 20 % Loss of prestress.
This is conservative for ordinary concrete (Walraven 1997) and thus for UHPC

Values that will be used in the design algorithm:

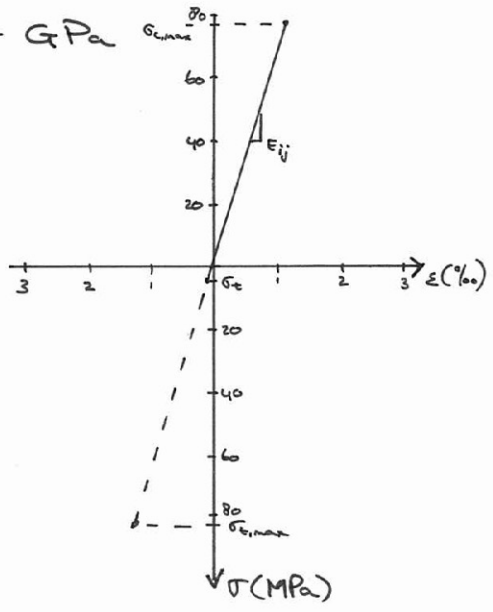
$\sigma_{c,max} = 0.75 \cdot 102 = -76.5$ MPa

$\sigma_t = f_{tj}/\gamma_{pf} = 4.6$ MPa

$\sigma_{t,max} = 0.80 \cdot 102 = 81.6$ MPa

The maximum tensile stress is only obtainable when pre-stress is applied. This involves a cost and will reduce the allowable compressive stress.

$E_{ij} = 65.000$ N/mm^2 $= 65$ GPa



B.2 B65 properties

CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. MT Afstuderen.	blz. no.
		berek. no. B65 stress-strain.	

NEN 6720 - blz 45

6720 - 6.1.1.

$f'_b = \frac{f'_{brep}}{\gamma_m} = \frac{0.72}{1.2} \cdot f'_{ck} = 39 \ N/mm^2$

$f'_{brep} = 0.72 \cdot f'_{ck}$

$\gamma_m = 1.2$

6720 - 6.1.2.

$f_b = \frac{f_{brep}}{\gamma_m} = \frac{3.01}{1.04} = 2.15 \ N/mm^2$

$f_{brep} = 0.7 \cdot (1.05 + 0.05 \cdot f'_{ck}) = 3.01$

$\gamma_m = 1.4$

6720 - 6.1.3

$E'_b = 22{,}250 + 250 f'_{ck}$

$= 38{,}500 \ N/mm^2$

6720 - 8.1.7 (blz 149)

maximale aanvangsdrukspanning

$\sigma'_{bi} \leq f'_{bt} * 0.75 f'_b$

$\sigma'_{bi} \leq 29.25 \ N/mm^2$

Voorspanverliezen (aanname) : 15 %

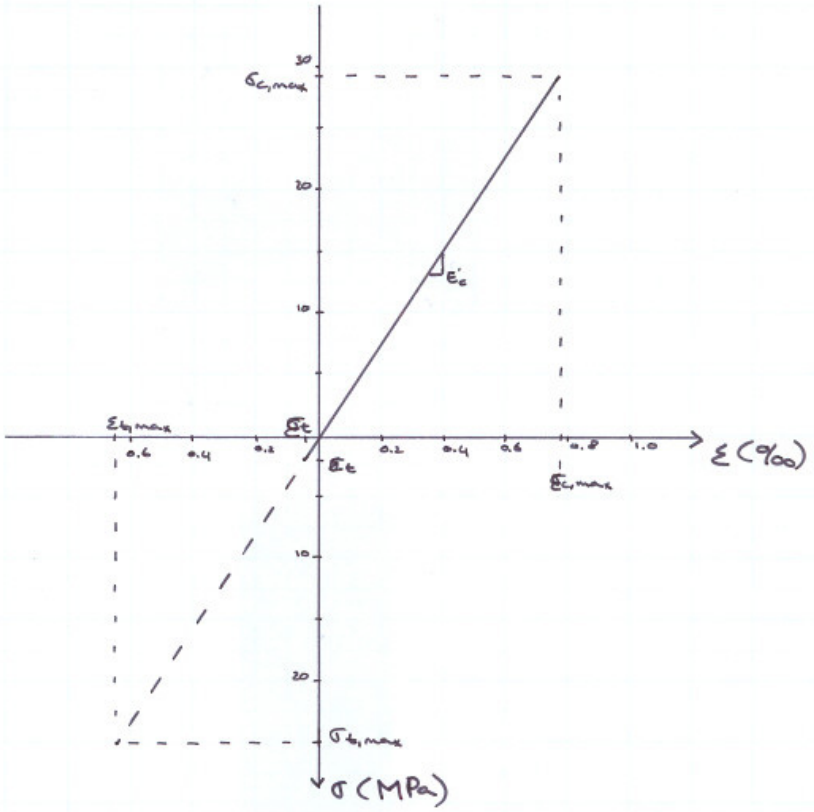
$\sigma'_b = (1-0.15) \cdot 29.25 = 25 \ N/mm^2$

Toelaatbare drukspanning

knikreductie (aanname): 25%

$\sigma'_b = (1-0.25) \cdot 39 = 29.3 \ N/mm^2$

CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. MF Afstuderen berek. no. BGS stress-strain	blz. no.



$\sigma_{c,max} = 0.75 \cdot 39 = -29.3 \ N/mm^2$

$\sigma_t = 2.15 \ N/mm^2$

$\sigma_{t,max} = 25 \ N/mm^2$

$E'_c = 38500 \ N/mm^2$

CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. MF Afstanderen	blz. no.
		berek. no. B65 8 Loads	

Calculation 080505 self weight.xls

Self Weight : 12.28 kN/m²

Dead Load : 0.80 kN/m²

Variable load : 0.56 kN/m² ↓ snow
0.95 kN/m² ↑ wind

SLS 1 : 12.28 + 0.80 + 0.56 = 13.64 kN/m²

SLS 2 : 12.28 + 0.80 − 0.95 = 12.13 kN/m²

ULS 1 : 1.2 · (12.28 + 0.8) + 1.5 · 0.56 = 16.54 kN/m²

ULS 2 : 1.2 · (12.28 + 0.8) + 1.5 · (−.95) = 14.27 kN/m²

B.3 Pre-stress or passive reinforcement trade-off

Passive reinforcement is maximally 8% when axially loaded columns are considered. The same maximum is assumed for this case. In the case of passively reinforced ultra-high performance concrete, the concrete will crack when the elastic tensile strength is applied. Now the reinforcement will solely handle the tensile stresses, up until the point when the reinforcement yields. The stress occurring at that point is the maximal stress that can be dealt with in the concrete.

$$\sigma_{u;d} = 0.08 \cdot 435 = 35MPa$$

The modulus of elasticity that comes with this cracked situation fully depends on the steel:

$$E'_c = 0.08 \cdot 200000 = 16000MPa$$

According to the applicable codes [6], the maximally applicable pre-stress for concrete is:

$$\sigma'_{ci} = 0.75 \cdot f'_b = 0.75 \cdot 0.6 \cdot 180 = 81MPa$$

Tensile stresses up to 81 MPa can be coped with without cracks developing in the concrete if the concrete is pre-stressed. The elastic modulus of elasticity now applies:

$$E'_c = 65000MPa$$

When pre-stressing is used, the allowable tensile stress is about twice as large and the modulus of elasticity about four times as large in comparison with passive reinforcement.

B.4 Pre-stress losses

Determination of pre-stress losses.

Calculations based on NEN6720 and Walraven 1997.

Data: Steel FeP1860: $f_{purep} = 1860$ MPa
$f_{pu} = 1690$ MPa
$\sigma_{po} = 1465$ MPa
$\sigma_{pi} = 1350$ MPa
$E_p = 200.000$ MPa

UHPC cross-section (assume 250×200 mm): $A = 50.000$ mm^2
$\emptyset = 900$ mm
$E_c = 65.000$ MPa
$\varepsilon_t = 700 \cdot 10^{-6}$
$\phi = 0{,}8$

Losses occur due to:
1) creep and shrinkage
2) relaxation of steel
3) anchorage setting
4) friction.

Ad 1)

$$\Delta\sigma_{ps} = \frac{\sigma_{cp}(0)\cdot n\cdot\phi + \varepsilon_d\cdot E_p}{1+n\cdot\omega_f\cdot(1+0{,}8\phi)}$$ (Walraven 1997)

$$n = \frac{E_c}{E_p} = \frac{65000}{200000} = 3{,}1$$

ε_d = that part of the shrinkage that occurs from the moment the pre-stress is applied onwards. Assume the pre-stress is applied after 28 days.

NEN 6720 gives: $k_t = \frac{t}{t+9{,}004\sqrt{h_m^3}}$

$h_m = 2\frac{A}{O}$

$h_m = 111$ mm

$k_t = \frac{t}{t+4{,}7}$

at $t=\infty$: $k_t = 1$

$t=28$: $k_t = 0{,}86$

$0{,}14$

14 % of the shrinkage occurs after day 28

$\varepsilon_d = 0{,}14\cdot 700\cdot 10^{-6} = 0{,}10$ ‰

σ_{cp} = stress in the concrete when the pre-stress is applied.

The concrete has the maximal pre-stress which is: ~~Fout~~ $\sigma_{u;d} = 8_1$ MPa.

The stress due to dead load creates a tensile stress in the pre-stressed elements. This "tensile" stress reduces the compressive stress that occurs in the element after pre-stressing.

$\sigma_{Dead\ load} = \frac{D.L.}{uls}\cdot 8_1 = \frac{1{,}54}{2{,}69} = 0{,}57\cdot 8_1 = 46$ MPa

resulting stress in concrete: $8_1 - 46 = 35$ MPa

$\sigma_{cp} = 35$ MPa

$f = 1$ (pre-stress is assumed centrical)

$$\omega = \frac{81}{1350} = 0{,}06$$

$$\Delta\sigma_{ps} = \frac{35 \cdot 3{,}1 \cdot 0{,}8 + 0{,}10 \cdot 10^{-3} \cdot 200000}{1 + 3{,}1 \cdot 0{,}06 \cdot 1 \cdot (1 + 0{,}8 \cdot 0{,}8)}$$

$$\Delta\sigma_{ps} = \frac{87 + 20}{1{,}3} = 82 \text{ MPa}$$

Ad 2)

$$\Delta\sigma_{pr} = 3 \cdot \Delta\sigma_{prel} \cdot \left(1 - 2\,\frac{\Delta\sigma_{pkr}}{\sigma_{pi}}\right) \qquad \text{(Walraven 1997)}$$

$$\Delta\sigma_{prel} = \frac{3}{100} \cdot \sigma_{pi}$$

$$\Delta\sigma_{pr} = 3 \cdot \frac{3}{100} \cdot 1350 \cdot \left(1 - 2\,\frac{82}{1350}\right)$$

$$\Delta\sigma_{pr} = 122 \cdot (1 - 0{,}12)$$

$$\Delta\sigma_{pr} = 107 \text{ MPa}$$

Ad 3)

Anchorage sets can be compensated for by overstressing during the pre-stressing process. According to NEN6720, the steel can be overstressed to 1465 MPa

Ad 4)

$F_p(x) = F_p \cdot e^{-\mu(\varphi+\varphi_1 x)}$ (Walraven 1997)

$\mu = 0{,}20$ (assumption based on Walraven 1997)

$\varphi = 0$ (straight pre-stressing tendons)

$\varphi_1 = 0{,}01$ (Wobble factor)

$x = 15\ m$ (half the length of the biggest element)

$F_p(x) = F_p \cdot e^{-0{,}20\,(0+0{,}01\cdot 15)}$

$F_p(15\,m) = F_p \cdot 0{,}97$

Resulting loss

$\Delta\sigma_p = 82 + 107 + (1-0{,}97)\cdot 1350 = \underline{230\ MPa}$

the loss is $\frac{230}{1350} = 17\ \%$

APPENDIX C ALGORITHM SOURCE CODE

```
      PROGRAM LINSTA

**LINSTA*************************************************************
* V4.01 21 April 2008 by Martin Flint                                 *
* The routine selects elements for a structure probablistically       *
* and assigns material to those selected. Then it performs a linear   *
* elastic analysis and calculates its performance.                    *
* It generates a performance value for each element over the best few *
* solutions that it had generated in the iterative step.              *
* This routine is used for the topruns on Omnisport_Coarse            *
*                                                                     *
* To be changed by the user:                                          *
*    - a few output statements according to number of elements        *
*    - number of comparisons in elitist ant determination             *
*    - the .dat file should have at least K+11 sets                   *
*    - the load applied in the .dat file should be 1 kN/m2 downward   *
*    - the elements location matrix                                   *
*                                                                     *
* The number of iterations, population size and elements are set in the *
* initialisation.                                                     *
* II is number of elements + 12                                       *
* JJ is number of ants + 5                                            *
* JE is number of elitist ants                                        *
* KK is number of iterations                                          *
* LL is number of nodes                                               *
* D  is number of dimensions                                          *
* DD is (#of dimensions-1)*3                                          *
**********************************************************************

      LOGICAL           EXIST, SDUM, HCK1, HCK2, HCK3, HCK4
      INTEGER           SFDISP, ASSGNF
      INTEGER           D, DD, I, II, J, JJ, JE, K, KK, L, LL, M, N
      INTEGER           NITS, NSLS, NSLE, NELS, NNOD, ELS, INTPT
      INTEGER           SELPNR(2), XSEC1, XSEC2, INFNR, HFAL
      INTEGER           DNODCR, ROWCNT, ROWMAX, NEIGHB(8)
      INTEGER           CONNEC, COLCNT, COLMAX
      INTEGER           HCCHCK, HRCHCK, HROW, HCOL, HOLE, HNR
      DOUBLE PRECISION X, DRRAND, RHO, LSLS1, LSLS2, LULS1, LULS2
      DOUBLE PRECISION RVINIT, RVMAX, RVMIN
      DOUBLE PRECISION PERF, PDELTA, PTR
      DOUBLE PRECISION DRES, DMX1, DMX2, DALLOW, DISPLA(3), DMAX
      DOUBLE PRECISION SVNMIS, SCALAR, SFYMN, SFYMX, SFT, STTRS2, STTRP2
```

```
      DOUBLE PRECISION SIGMX1, SIGMN1, SIGMX2, SIGMN2, STTRS1, STTRP1
      DOUBLE PRECISION SRANG1, SRANG2, SRANGE
      DOUBLE PRECISION MSTRUC, MELEM, VOL, VPTOT, VPLEV, HOLES
      DOUBLE PRECISION EWIDTH, EHEIGH, ETHICK, EVOL
      CHARACTER*24     XSTRTM
      CHARACTER*17     XUSEL1
      CHARACTER*5      XUSEL2
      PARAMETER        (LL=903,KK=500,JJ=105,JE=8,II=852,DD=3,D=2)
      PARAMETER        (ROWMAX=20,COLMAX=42)
      DOUBLE PRECISION STRESS(DD), SPRI(3), SPAXES(D*D), STRINT(DD)
      DOUBLE PRECISION STMIMA(3,JJ-4,II-8), DATSTR(JJ,II)
      INTEGER          E(JE), INFO(II-8)
      INTEGER          NODES((D-1)*4), ELEMTS(0:(COLMAX+1),0:(ROWMAX+1))
      LOGICAL          NODUSE(LL), HNOMIN(COLMAX,ROWMAX)
      CALL APPBEG( 'LINSTA' )

C     Specify initial probability and initialise random function
      OPEN (UNIT=1, FILE='structures.tb', STATUS='NEW')
      OPEN (UNIT=2, FILE='jobinfo.tb'   , STATUS='NEW')
      OPEN (UNIT=3, FILE='pheromone.tb' , STATUS='NEW')
      OPEN (UNIT=4, FILE='elemts.tb'    , STATUS='NEW')
      OPEN (UNIT=5, FILE='performance.tb',STATUS='NEW')

C     Determine the constants that will be used throughout
      NELS    = II-12
      NSLS    = JJ-5
      NSLE    = JE
      NITS    = KK
      NNOD    = LL
      RVINIT  = 0.80
      RVMAX   = 1.00
      RVMIN   = 0.00
      RHO     = 0.20
      SFYMN   = -765E5
      SFYMX   = 816E5
      SFT     =  46E5
      DALLOW  = 0.500
      INTPT   = 4
      SCALAR  = 1/(1.0*INTPT)
      STTRP1  = 0.90
      STTRS1  = 0.01
      STTRP2  = 0.90
      STTRS2  = 0.05
      LULS1   = 3.00
      LULS2   = 0.20
      LSLS1   = 2.26
      LSLS2   = 0.85
      ETHICK  = 0.100
      EWIDTH  = 1.250
      EHEIGH  = 0.500
      HROW    = 4
      HCOL    = 1
      HNR     = 2

C     Check if parameters D and DD are consistent
      L = (D-1)*3
      IF (L .NE. DD) CALL PRGERR('LINSTA', 002)

      WRITE(1,FMT='(A4, A5, A7, 8A11, A7, A11, A5)') 'ITER','STR','VOL'
     + ,'DMAX(1)','DMAX(2)','SIGMAX(1)','SGMXIN(1)','SIGMAX(2)'
```

```
     + ,'SIGMIN(2)','SURFACE','PRESTPROD','HOLES'
     + ,'PERF','RANK'
      DO 10 I = 1, NELS
         DATSTR((NSLS+2),I)=RVINIT
  10  CONTINUE

      EVOL = EWIDTH*EHEIGH*ETHICK
      CALL GTCH('/JOBINF/RUN/STRTIM', XSTRTM)
      READ(XSTRTM,'(A17,2I1,A5)') XUSEL1, XSEC1, XSEC2, XUSEL2

C PARAM Create the elements matrix
      WRITE(4,FMT='(44I4)') (ELEMTS(J,0),J=0,43)
      DO 30 I = 1, ROWMAX
         DO 20 J = 1, COLMAX
            ELEMTS(J,I) = 42*(20-I) + J
  20     CONTINUE
         WRITE(4,FMT='(44I4)') (ELEMTS(J,I),J=0,43)
  30  CONTINUE
      WRITE(4,FMT='(44I4)') (ELEMTS(J,21),J=0,43)

C     Print the job info
      WRITE(2,FMT='(A24, I6)')   'Dimensions '             , D
      WRITE(2,FMT='(A24, I6)')   'Stress components '      , DD
      WRITE(2,FMT='(A24, I6)')   'Nodes '                  , NNOD
      WRITE(2,FMT='(A24, I6)')   'Elements '               , NELS
      WRITE(2,FMT='(A24, I6)')   'Integration points '     , INTPT
      WRITE(2,FMT='(A24, F6.3)') 'Element width '          , EWIDTH
      WRITE(2,FMT='(A24, F6.3)') 'Element height '         , EHEIGH
      WRITE(2,FMT='(A24, F6.3)') 'Element thickness '      , ETHICK
      WRITE(2,FMT='(A24, F6.3)') 'Volume per element '     , EVOL
      WRITE(2,FMT='(A24, F6.2)') 'Space height '       , (ROWMAX*EHEIGH)
      WRITE(2,FMT='(A24, F6.2)') 'Space width '        , (COLMAX*EWIDTH)
      WRITE(2,FMT='(A24, I6)')   'Iterations '             , NITS
      WRITE(2,FMT='(A24, I6)')   'Ants '                   , NSLS
      WRITE(2,FMT='(A24, I6)')   'Elitist ants '           , NSLE
      WRITE(2,FMT='(A24, I6)')   'Ant runs '               , (NSLS*NITS)
      WRITE(2,FMT='(A24, E11.3)')'Fy,max '                 , SFYMX
      WRITE(2,FMT='(A24, E11.3)')'Fy,min '                 , SFYMN
      WRITE(2,FMT='(A24, E11.3)')'Ft '                     , SFT
      WRITE(2,FMT='(A24, F6.3)') 'Allowed deflection '     , DALLOW
      WRITE(2,FMT='(A24, F6.3)') 'RV initial '             , RVINIT
      WRITE(2,FMT='(A24, F6.3)') 'RV maximum '             , RVMAX
      WRITE(2,FMT='(A24, F6.3)') 'RV minimum '             , RVMIN
      WRITE(2,FMT='(A24, F6.3)') 'Stress treshold 1 '      , STTRS1
      WRITE(2,FMT='(A24, F6.3)') 'Performance treshold 1 ' , STTRP1
      WRITE(2,FMT='(A24, F6.3)') 'Stress treshold 2 '      , STTRS2
      WRITE(2,FMT='(A24, F6.3)') 'Performance treshold 2 ' , STTRP2
      WRITE(2,FMT='(A24, F6.3)') 'Rho '                    , RHO
      WRITE(2,FMT='(A24, F6.2)') 'Load in ULS 1 '          , LULS1
      WRITE(2,FMT='(A24, F6.2)') 'Load in ULS 2 '          , LULS2
      WRITE(2,FMT='(A24, F6.2)') 'Load in SLS 1 '          , LSLS1
      WRITE(2,FMT='(A24, F6.2)') 'Load in SLS 2 '          , LSLS2
      WRITE(2,FMT='(A24, I6)')   'Preferred nr of holes '  , HNR
      WRITE(2,FMT='(A24, F6.2)') 'Hole height '          , (HROW*EHEIGH)
      WRITE(2,FMT='(A24, F6.2)') 'Hole width '           , (HCOL*EWIDTH)
      WRITE(2,FMT='(A24, A24)')  'Start time '             , XSTRTM


******PER ITERATION*****************************************************
C     Start the iterations
```

```
* Not available on web version
```

* Not available on web version

```
* Not available on web version
```

* Not available on web version

```
* Not available on web version
```

* Not available on web version

```
* Not available on web version
```

* Not available on web version

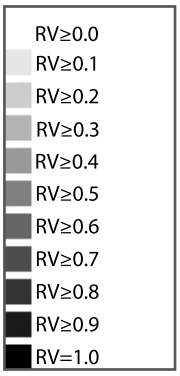
```
 100  CONTINUE
******PER ITERATION*****************************************************


      CALL DIANA( 'NEXT  ',  ' ', 0 )

      END
```

The code printed above is used for the runs on the coarse mesh of the Omnisport hall roof. For other runs, the used code may be slightly different.

APPENDIX D ALGORITHM RUNS

D.1 Validation runs

Validation runs are performed on a problem which has a Michell structure as the analytical optimum. The runs are split into two goups; runs on a coarse mesh and runs on a fine mesh. All visual outputs are mirrored along the line of symmetry; only half the problem is analysed. Plots that show a pheromone distribution show the reference value on the elements. The darker the element, the higher the probability that is is chosen as a part of the solution. The following gray-scale gives the explanation.



Greyscale for pheromone plots.

The coarse runs consist of five runs performed on the same problem using the same run parameters (runs 6-10). Run 11 performs more iterations, run 12 uses more strict boundary conditions and run 16 is subject to a bigger load. First general information is given after which each run is described seperately.

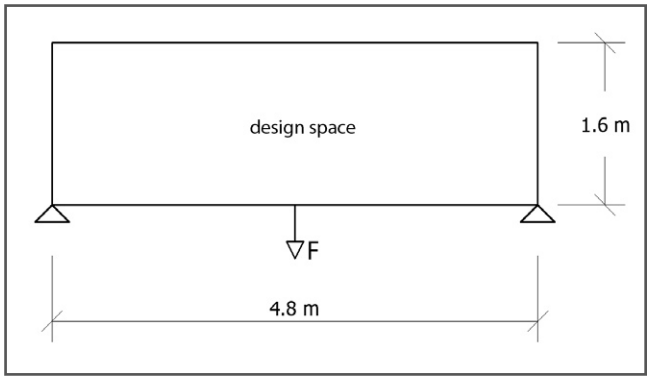
On the fine mesh, five runs on the same problem are presented.

Manual calculations are given for the validation of the DIANA calculations.

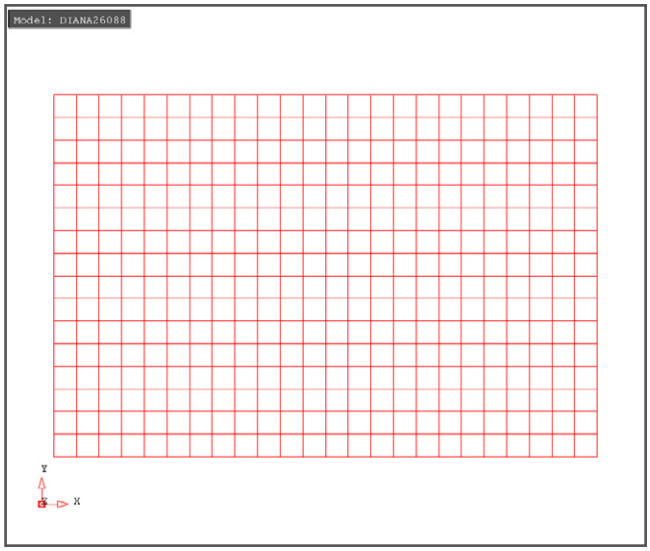
D.1.1 Coarse Michell truss

Dimensions	2	
Stress components	3	
Nodes	425	
Elements	384	
Integration points	4	
Elitist ants	10	
RV initial	1	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	5.00E+07	Pa
Fy;min	-5.00E+07	Pa
Ft	5.00E+07	Pa
Allowed deflection	0.015	m

Job information.



Design space and boundary conditions.



Mesh.

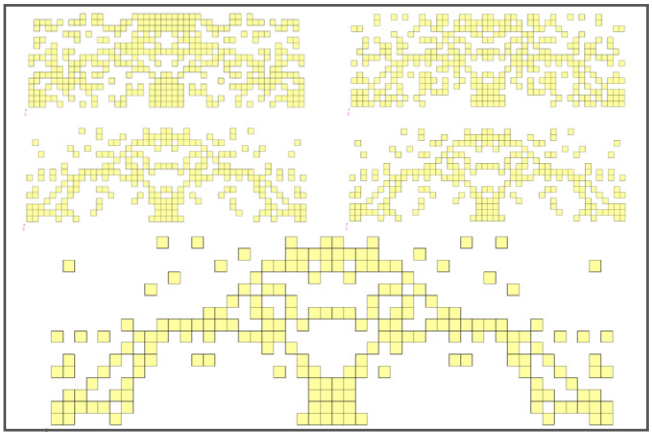
Run 6

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:51:54	hh:mm:ss
Run time	3114	s
Time per ant run	0.125	s
Load	400	kN

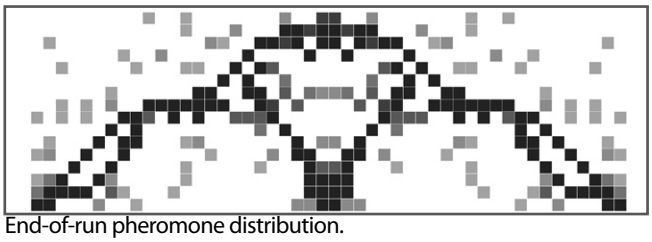
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

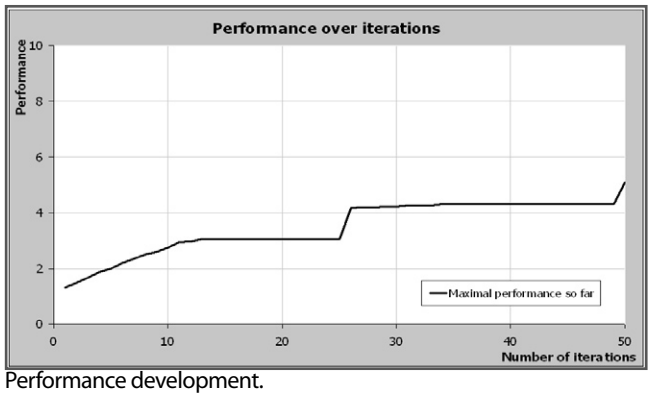
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

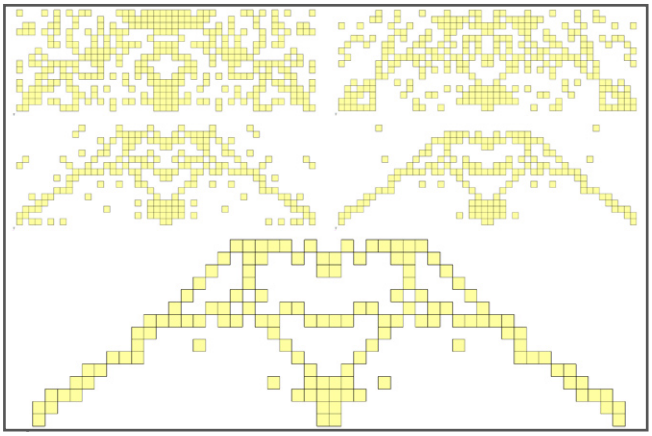
Run 7

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:52:27	hh:mm:ss
Run time	3147	s
Time per ant run	0.126	s
Load	400	kN

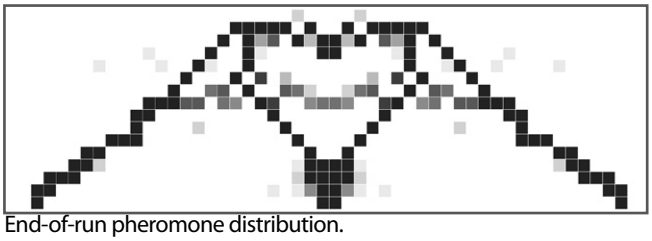
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

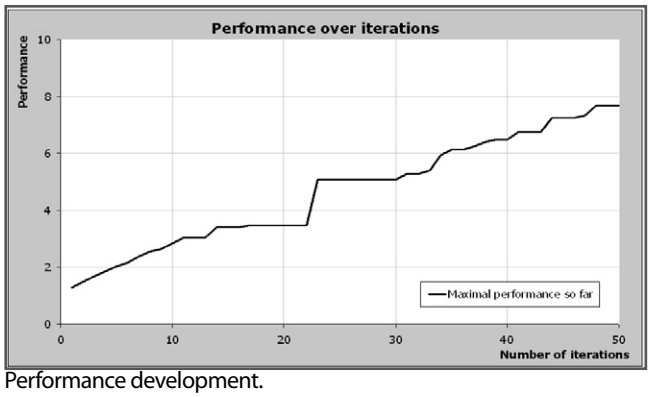
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

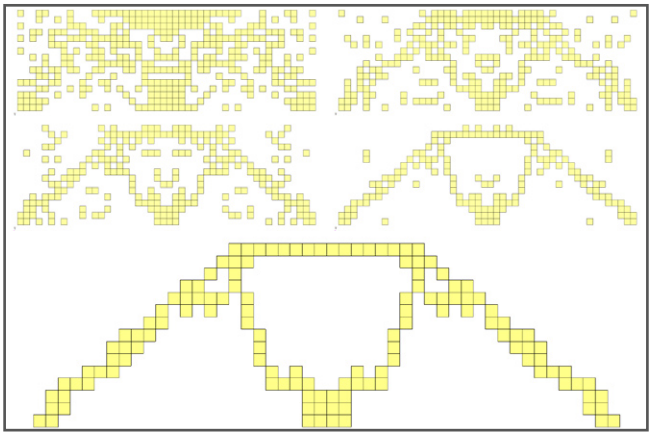
Run 8

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:46:35	hh:mm:ss
Run time	3795	s
Time per ant run	0.112	s
Load	400	kN

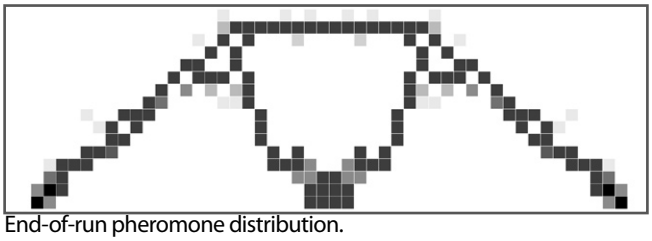
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

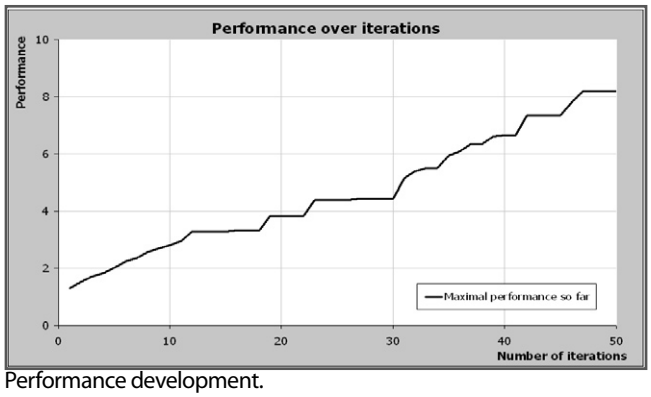
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

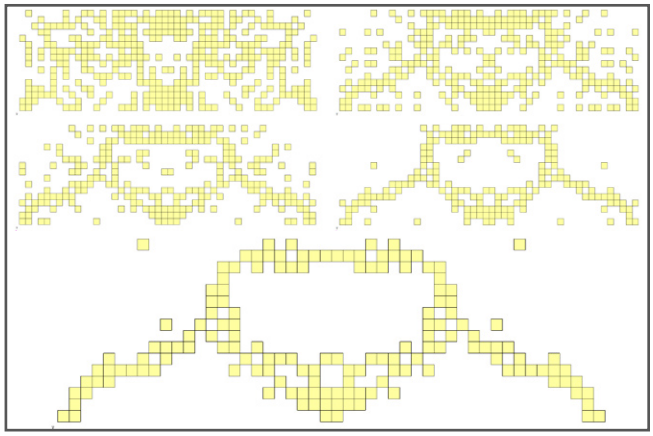
Run 9

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:48:44	hh:mm:ss
Run time	2924	s
Time per ant run	0.117	s
Load	400	kN

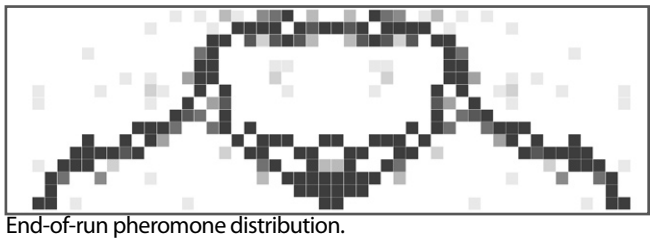
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

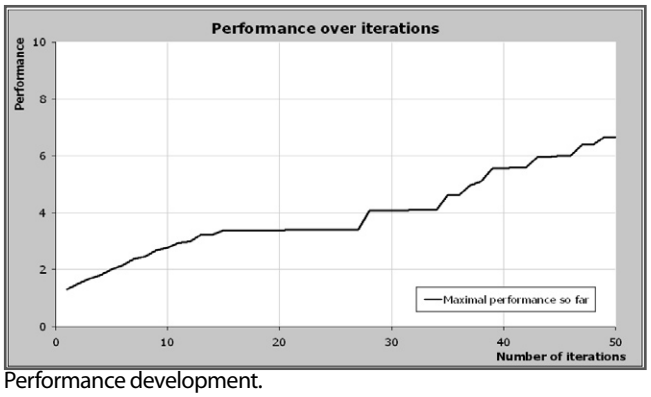
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

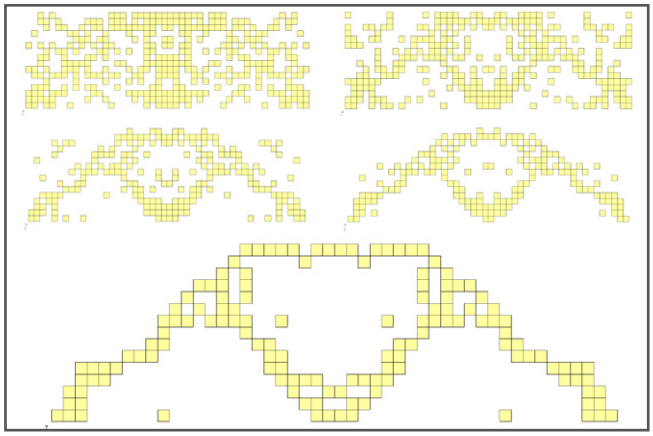
Run 10

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:46:53	hh:mm:ss
Run time	2813	s
Time per ant run	0.113	s
Load	400	kN

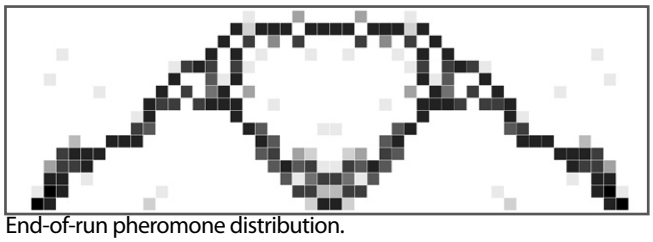
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

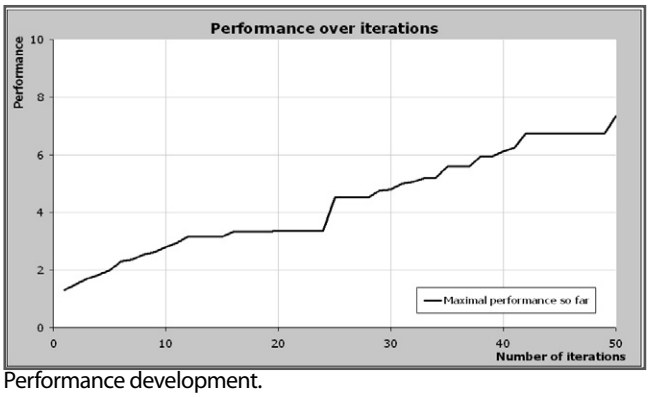
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

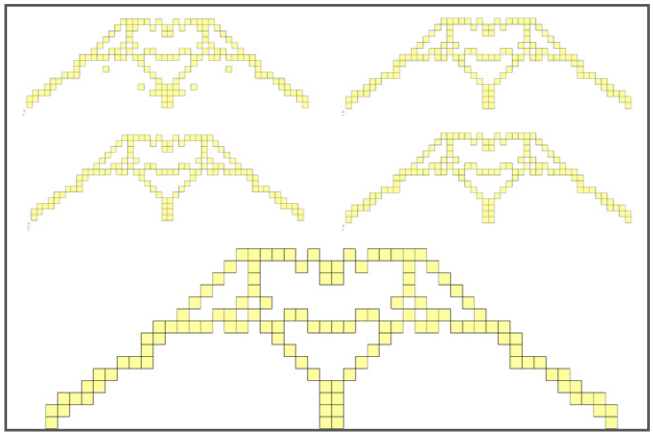
Run 11

Iterations	250	
Ants	500	
Elitist ants	10	
Ant runs	125000	
Run time	4:11:59	hh:mm:ss
Run time	15119	s
Time per ant run	0.121	s
Load	400	kN

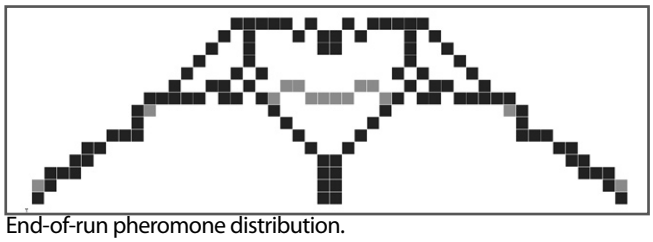
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ 0.1 f_y \le \sigma_i < 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 1.5 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

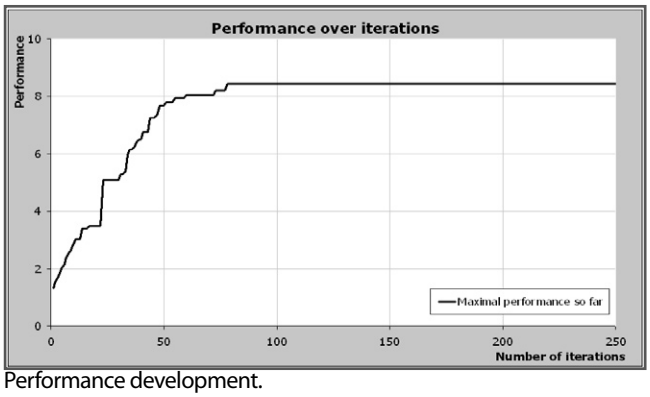
Specific job information and performance function.



Best solutions found after 50, 100, 150, 200 and 250 iterations.



End-of-run pheromone distribution.



Performance development.

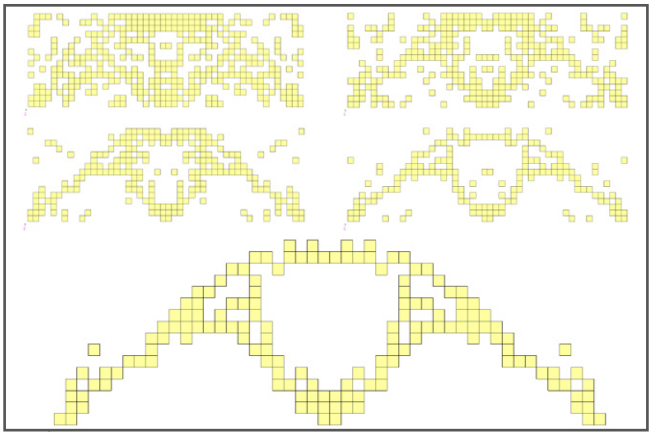
Run 12

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:45:09	hh:mm:ss
Run time	2709	s
Time per ant run	0.108	s
Load	400	kN

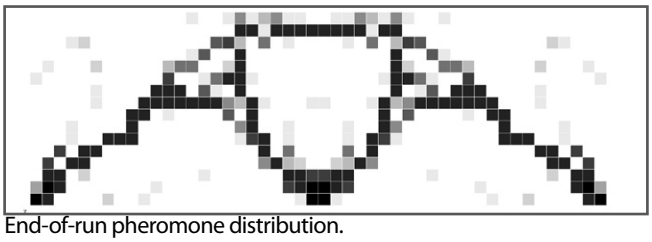
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-8} & if\ |\sigma|_{max} > f_y \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ \sigma_i \ge 0.1 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

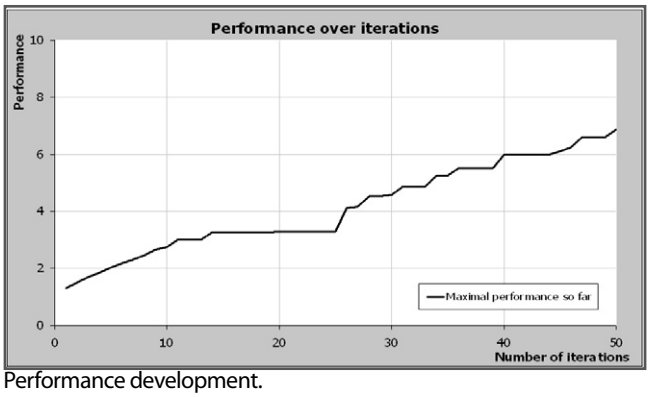
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

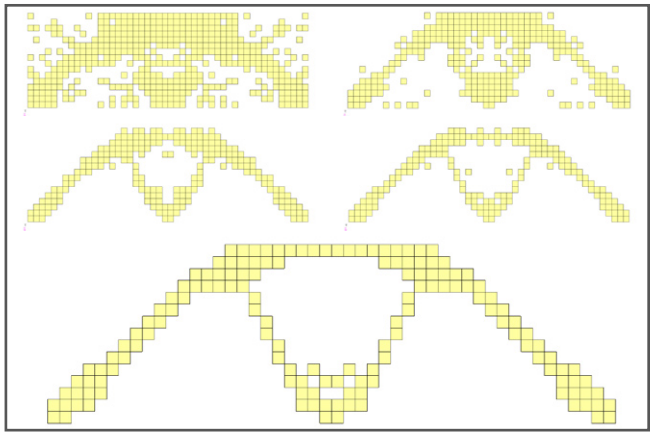
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Iterations	50	
Ants	500	
Elitist ants	10	
Ant runs	25000	
Run time	0:45:24	hh:mm:ss
Run time	2724	s
Time per ant run	0.109	s
Load	600	kN

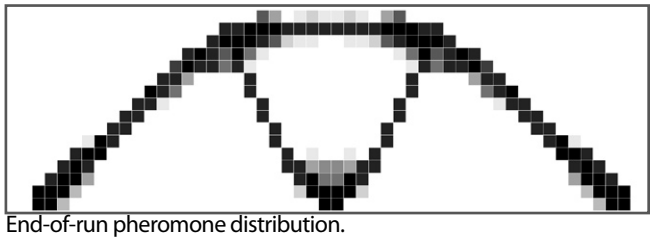
$$perf_j = \begin{cases} 1 \cdot 10^{-9} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-8} & if\ |\sigma|_{max} > f_y \\ \frac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ \sigma_i \geq 0.1 f_y \\ perf_{i,k} + 0.5 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.1 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

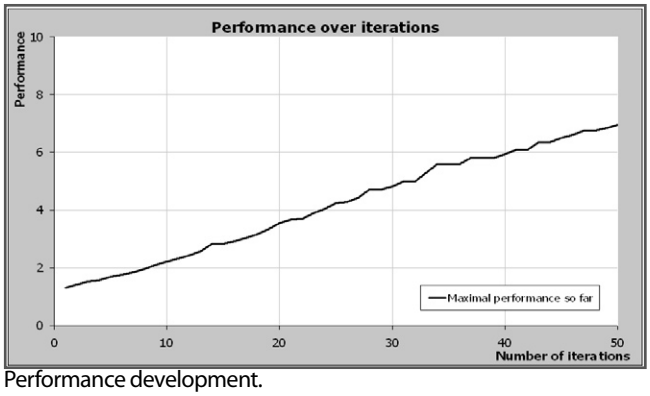
Specific job information and performance function.



Best solutions found after 10, 20, 30, 40 and 50 iterations.



End-of-run pheromone distribution.



Performance development.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

D.1.2 Fine Michell truss

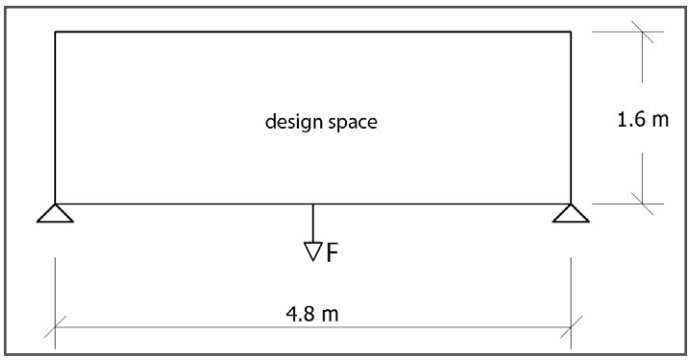
Dimensions	2	
Stress components	3	
Nodes	1617	
Elements	1536	
Integration points	4	
Iterations	250	
Ants	200	
Elitist ants	8	
Ant runs	50000	
RV initial	0.9	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	5.00E+07	Pa
Fy;min	-5.00E+07	Pa
Ft	5.00E+07	Pa
Allowed deflection	0.01	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load	600	kN

Job information.

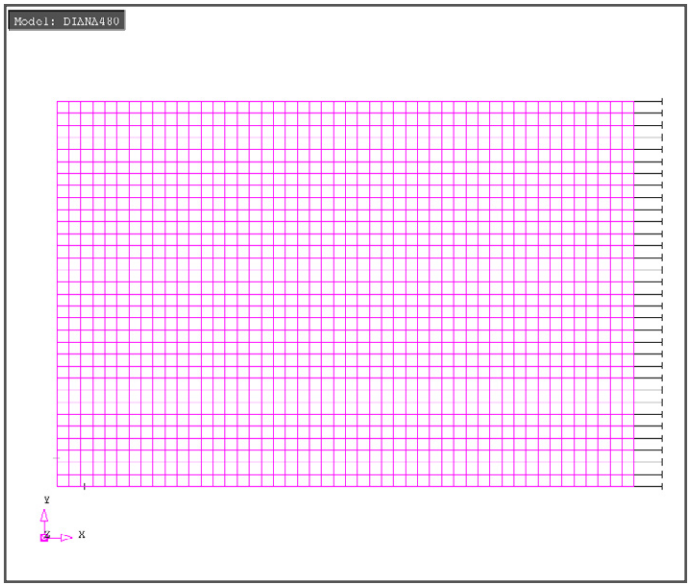
$$perf_j = \begin{cases} 1 \cdot 10^{-6} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-5} & if\ |\sigma|_{max} > f_y \\ \dfrac{1}{vol} & else \end{cases}$$

$$perf_{i,k} = \begin{cases} perf_{i,k} + perf_{j,k} & if\ i = material\ and\ \sigma_i \geq 0.05 f_y \\ perf_{i,k} + 0.9 \cdot perf_{j,k} & if\ i = material\ and\ \sigma_i < 0.05 f_y \\ perf_{i,k} & if\ i = void \end{cases}$$

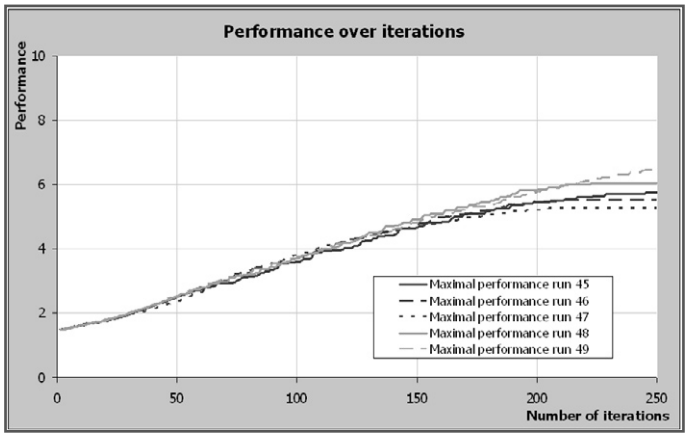
Performance function.



Design space and boundary conditions.



Mesh.

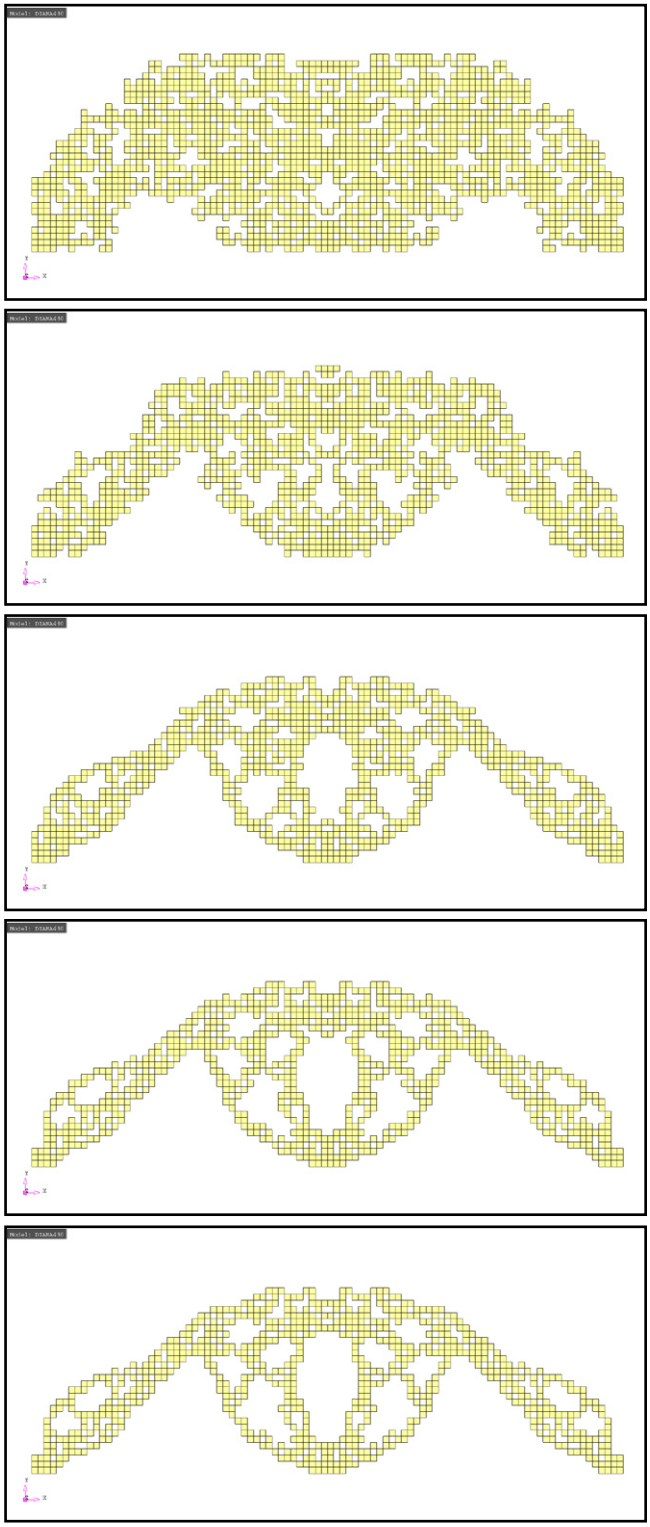


Performance development of the several runs.

Run 45

Run time	9:07:01	hh:mm:ss
Run time	32821	s
Time per ant run	0.656	s

Run times.

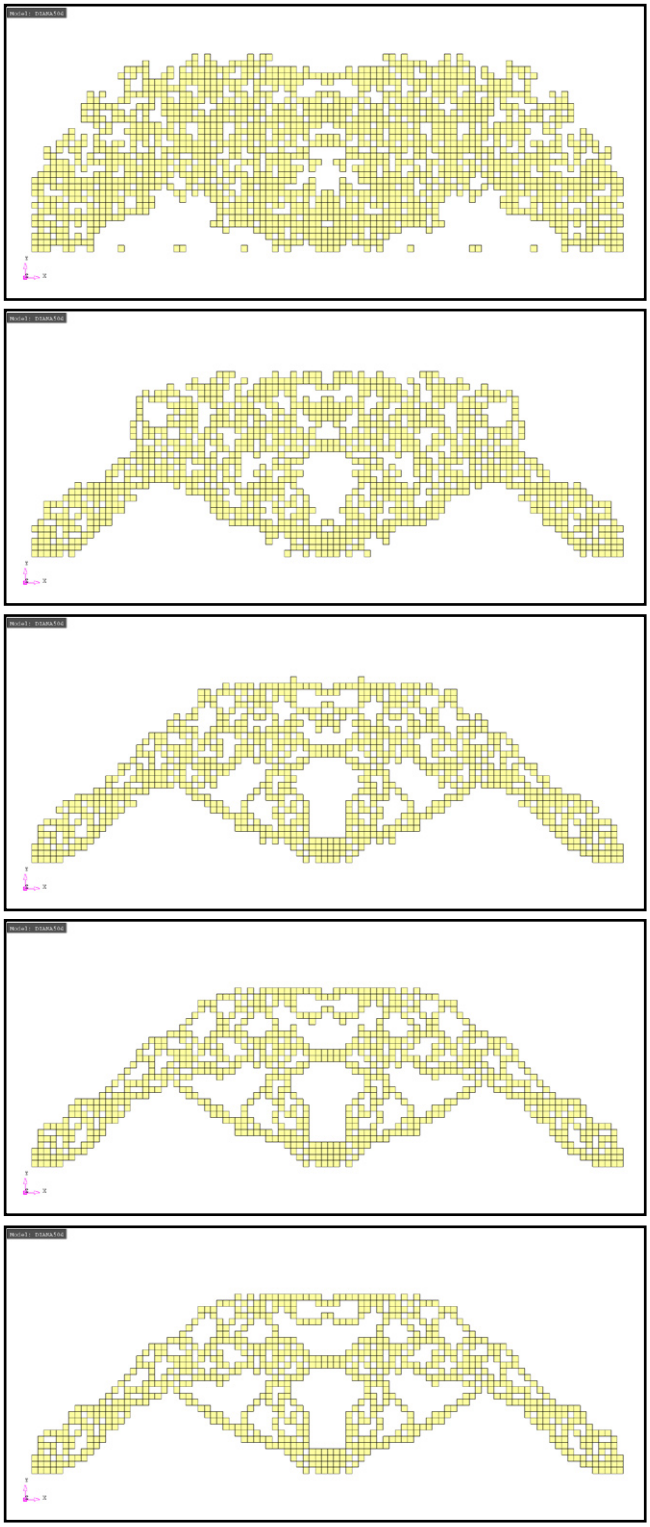


Best solutions found after 50, 100, 150, 200 and 250 iterations.

Run 46

Run time	9:10:43	hh:mm:ss
Run time	33043	s
Time per ant run	0.661	s

Run times.



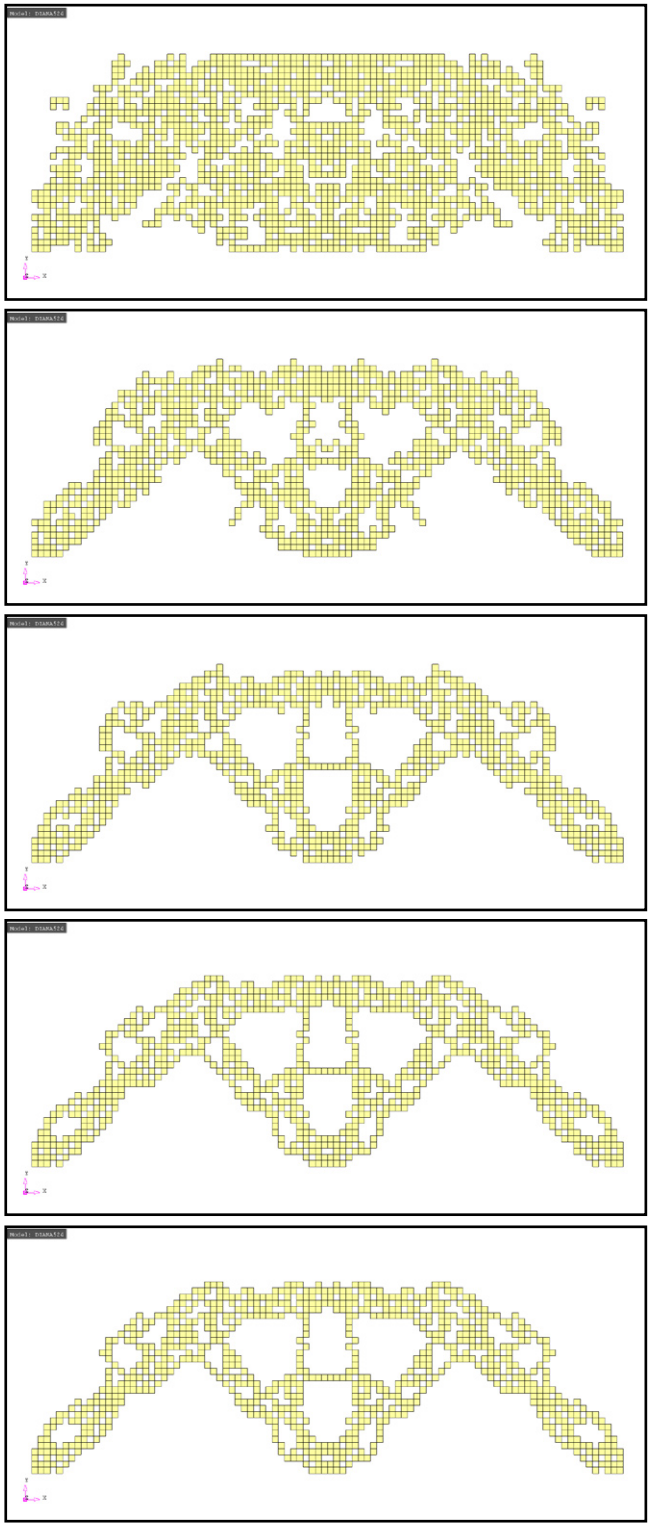
Best solutions found after 50, 100, 150, 200 and 250 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 47

Run time	9:12:14	hh:mm:ss
Run time	33134	s
Time per ant run	0.663	s

Run times.

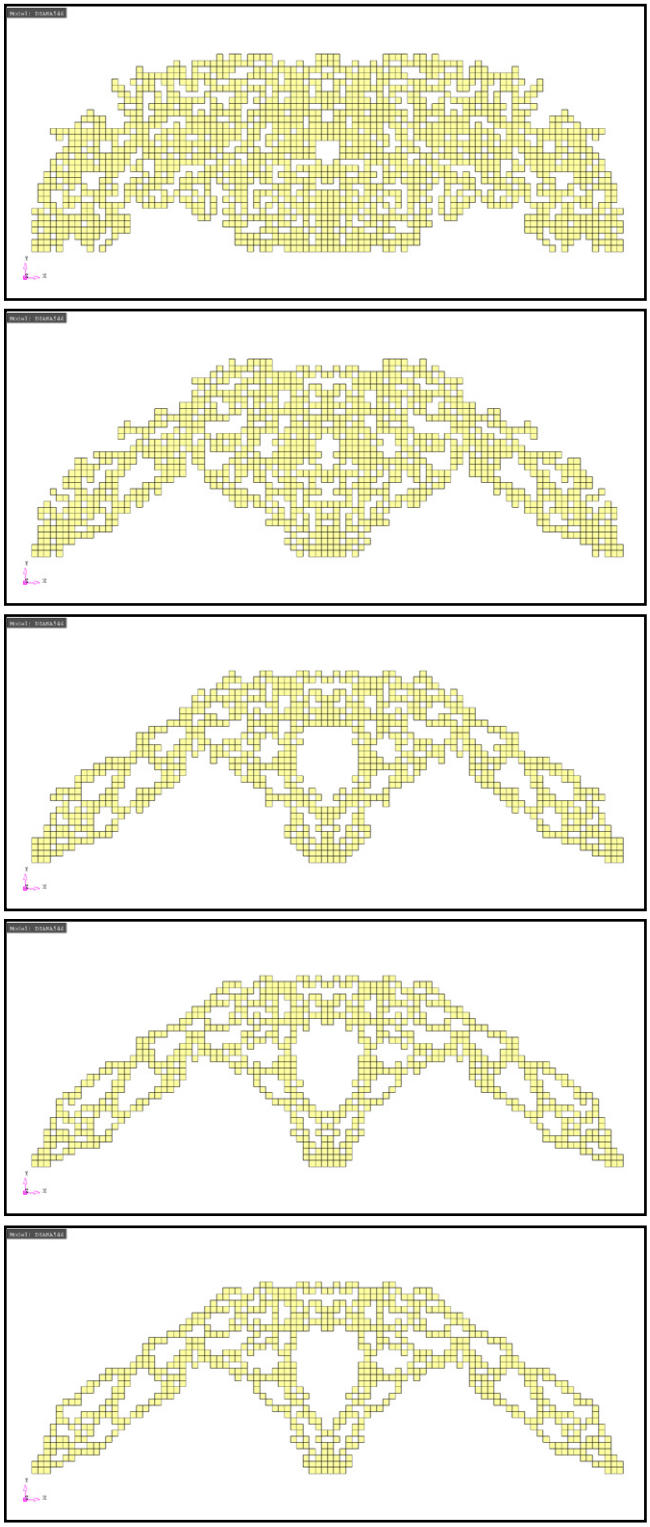


Best solutions found after 50, 100, 150, 200 and 250 iterations.

Run 48

Run time	9:04:33	hh:mm:ss
Run time	32673	s
Time per ant run	0.653	s

Run times.



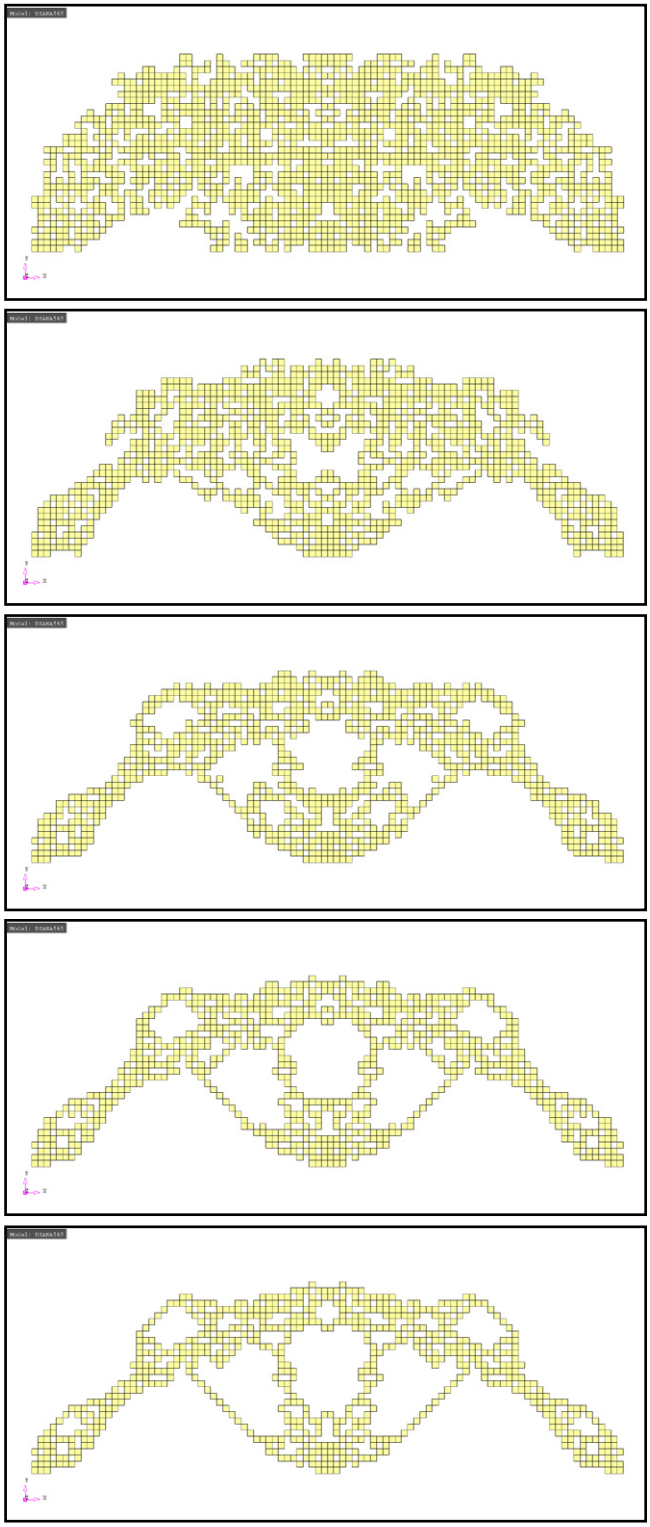
Best solutions found after 50, 100, 150, 200 and 250 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 49

Run time	9:10:10	hh:mm:ss
Run time	33010	s
Time per ant run	0.660	s

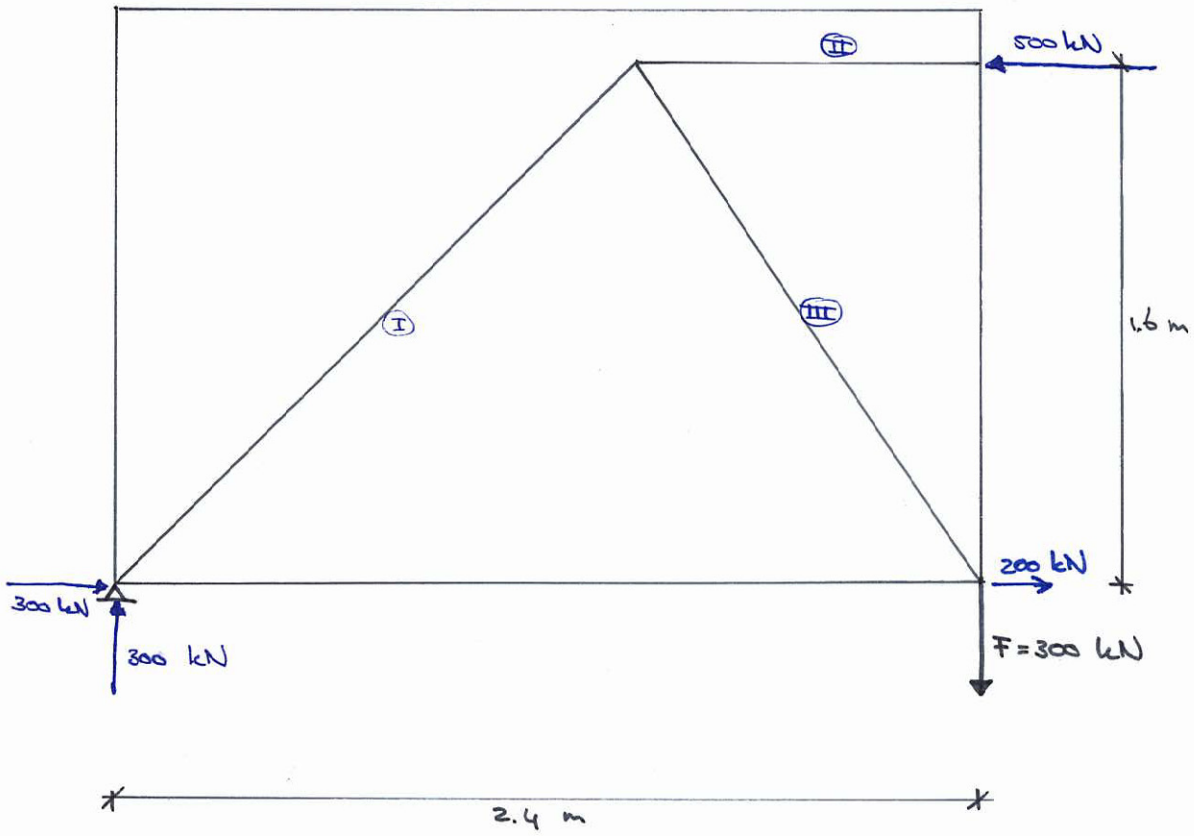
Run times.



Best solutions found after 50, 100, 150, 200 and 250 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. MF Afstudeer	blz. no.
		berek. no. handcalc coarse validation	



I : $N = -424$ kN $\rightarrow A_I \geq 8480$ mm^2 $= 100 \times 85$ mm

II : $N = -500$ kN $\rightarrow A_{II} \geq 10000$ mm^2 $= 100 \times 100$ mm

III : $N = +360$ kN $\rightarrow A_{III} \geq 7200$ mm^2 $= 100 \times 72$ mm

Allowed stress $= 50$ N/mm^2

D.2 Parameter sensitivity

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	250	
Ants	100	
Elitist ants	5	
Ant runs	25000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.00E+07	Pa
Fy;min	-8.00E+07	Pa
Ft	5.00E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.7	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.24	kN/m^2
Load case ULS2	0.98	kN/m^2
Load case SLS1	2.56	kN/m^2
Load case SLS2	1.05	kN/m^2
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m

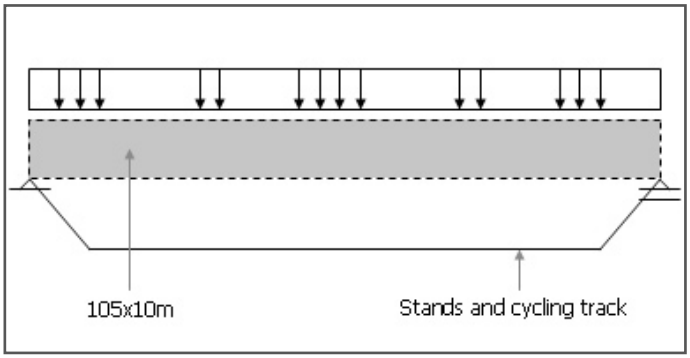
Basic job information. If not indicated, above values are used in the runs.

Run time	1:49:10	hh:mm:ss
Run time	6650	s
Time per ant run	0.262	s

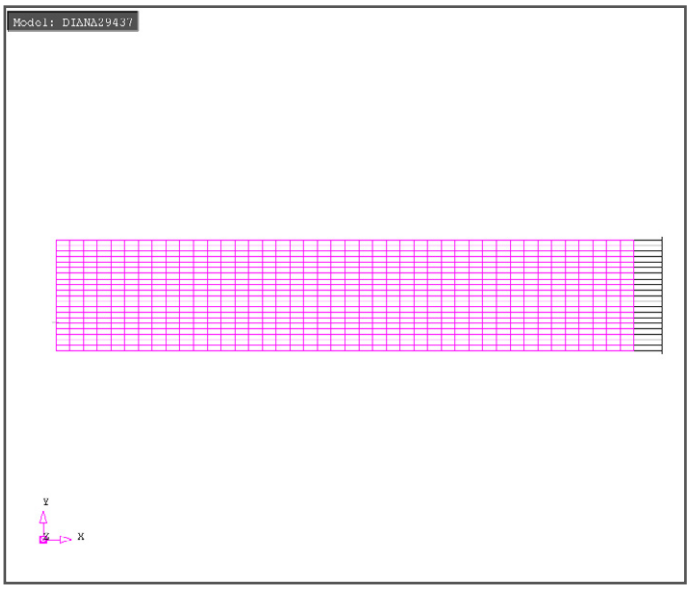
Average of the run times for runs with 25000 ant runs.

$$perf_j = \begin{cases} 1 \cdot 10^{-6} & if\ u_{max} > u_{allowable} \\ 1 \cdot 10^{-5} & if\ \sigma_{max} > \sigma_{allowable;max} \\ 1 \cdot 10^{-5} & if\ \sigma_{min} < \sigma_{allowable;min} \\ 2 \cdot 10^{-5} & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1}{2V + 0.3S + 0.5PV_p + 20H} & else \end{cases}$$

Performance function.



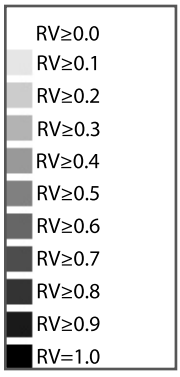
Design space and boundary conditions.



Mesh.

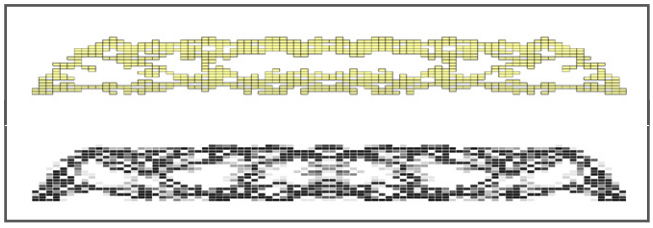
The parameter sensitivity runs are performed with the parameter set as described on this page as a basis. In each run, one of the parameters is altered. Four runs have been performed on each parameter set. All visual output is mirrored along the line of symmetry; only half the problem is analysed.

Wherever a visual representation of the pheromone distribution is given, the following legend applies.

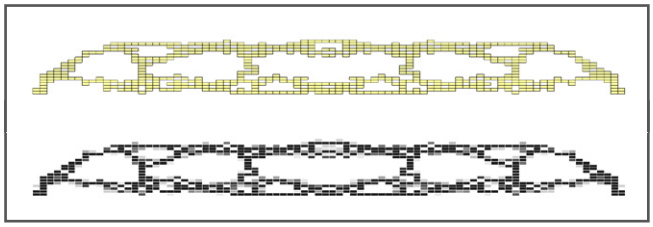


Greyscale for pheromone plots.

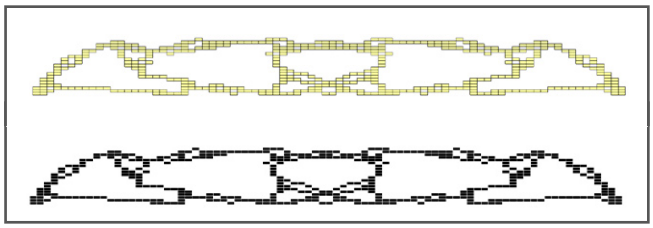
D.2.1 Pheromone decay



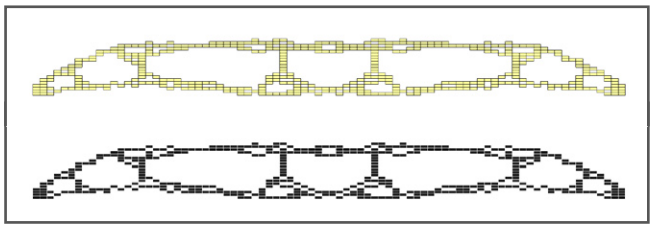
Best solution with rho = 0.05 and end-of-run pheromone distribution.



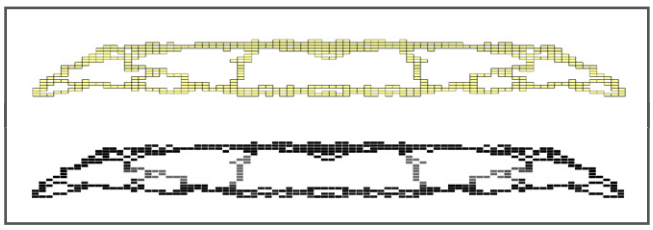
Best solution with rho = 0.10 and end-of-run pheromone distribution.



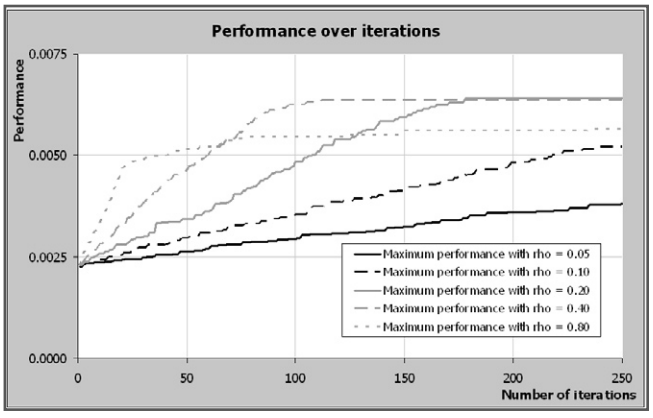
Best solution with rho = 0.20 and end-of-run pheromone distribution.



Best solution with rho = 0.40 and end-of-run pheromone distribution.

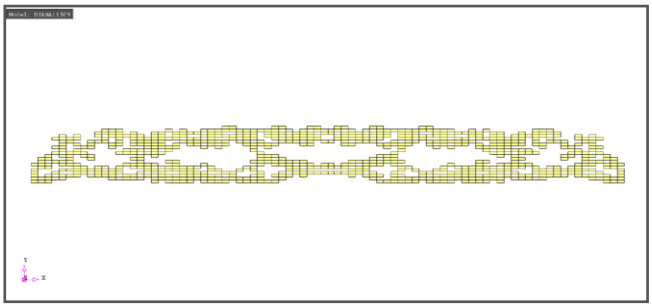


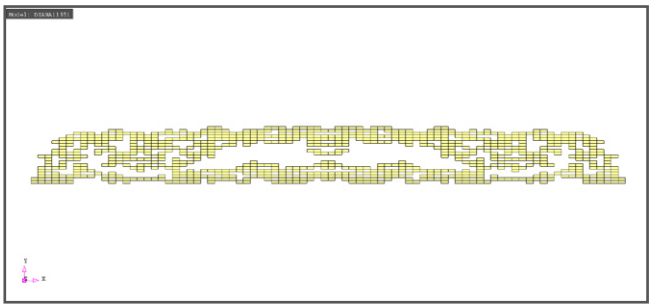
Best solution with rho = 0.80 and end-of-run pheromone distribution.

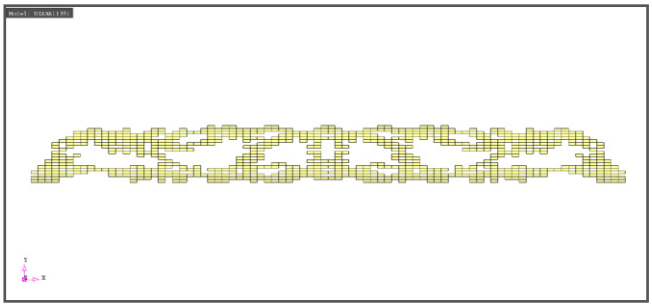


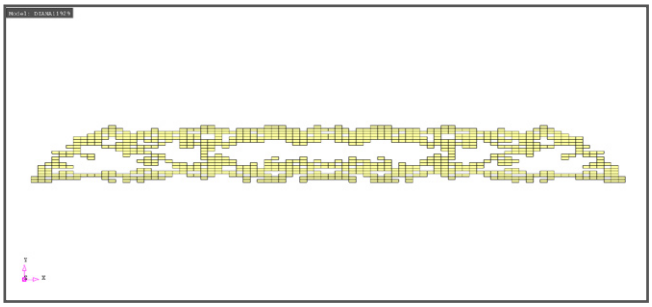
Influence of rho on the performance chart.

Rho = 0.05

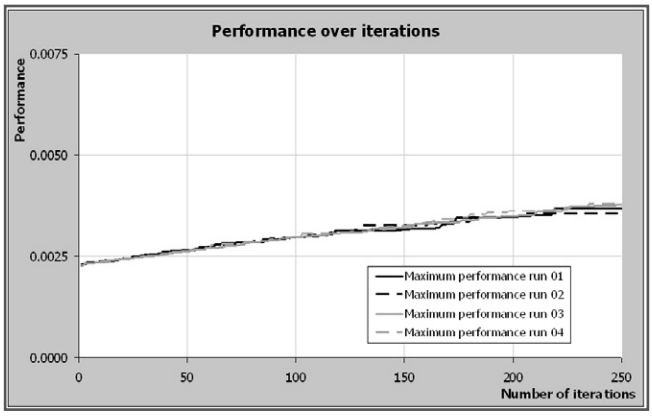






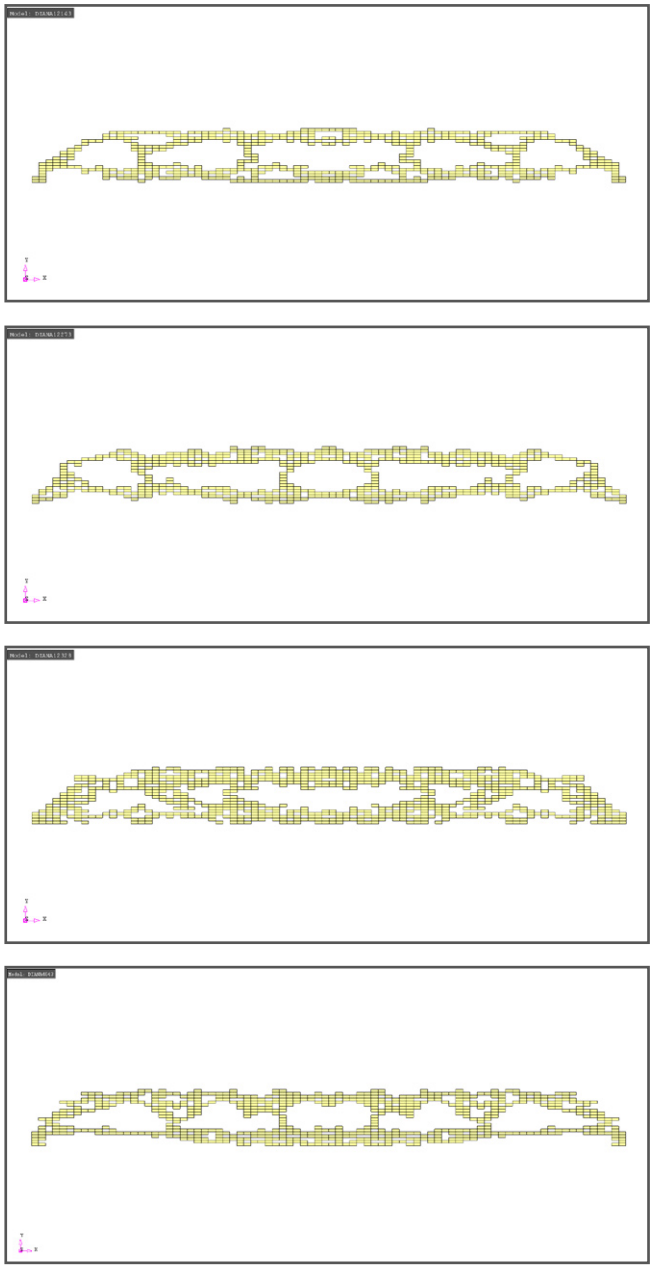


Four runs have been performed resulting in the above four different best solutions.



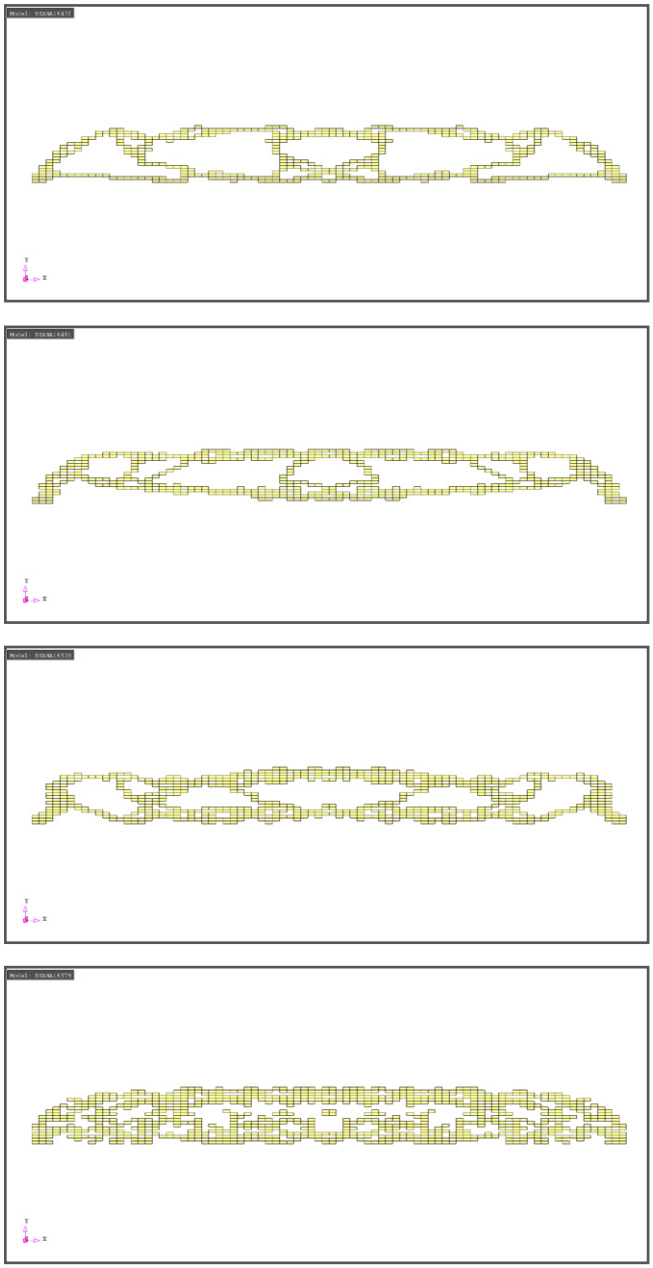
Performance of the four runs that have been performed.

Rho = 0.10

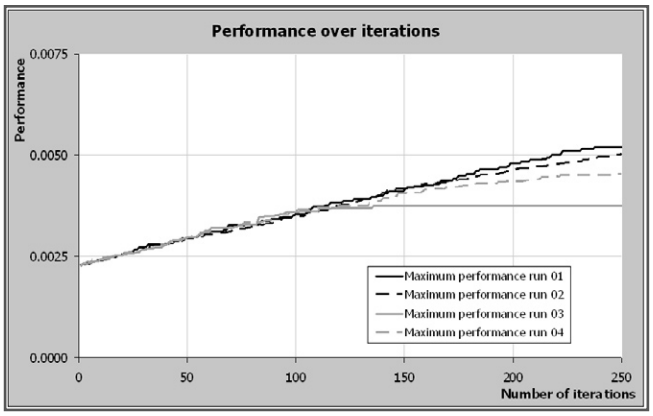


Four runs have been performed resulting in the above four different best solutions.

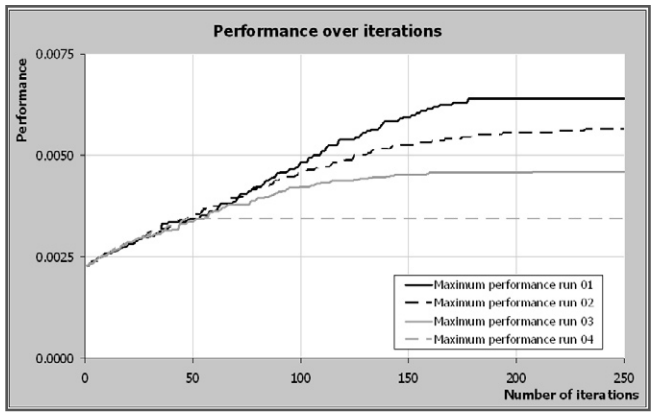
Rho = 0.20



Four runs have been performed resulting in the above four different best solutions.



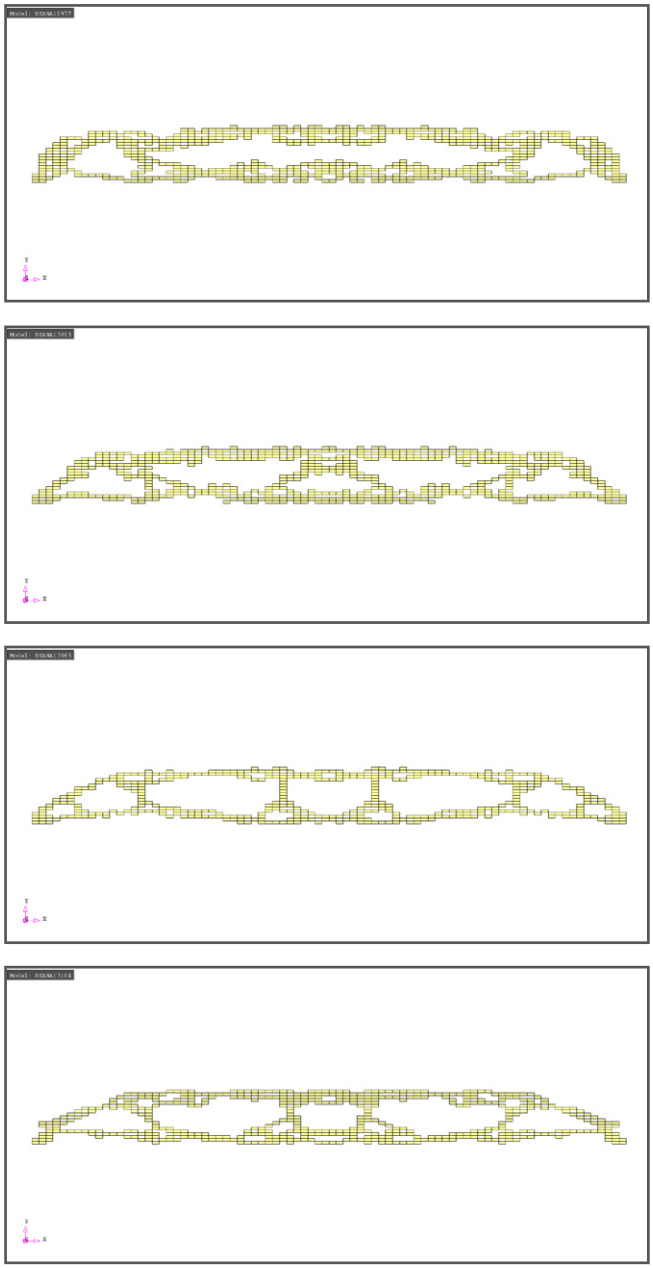
Performance of the four runs that have been performed.



Performance of the four runs that have been performed.

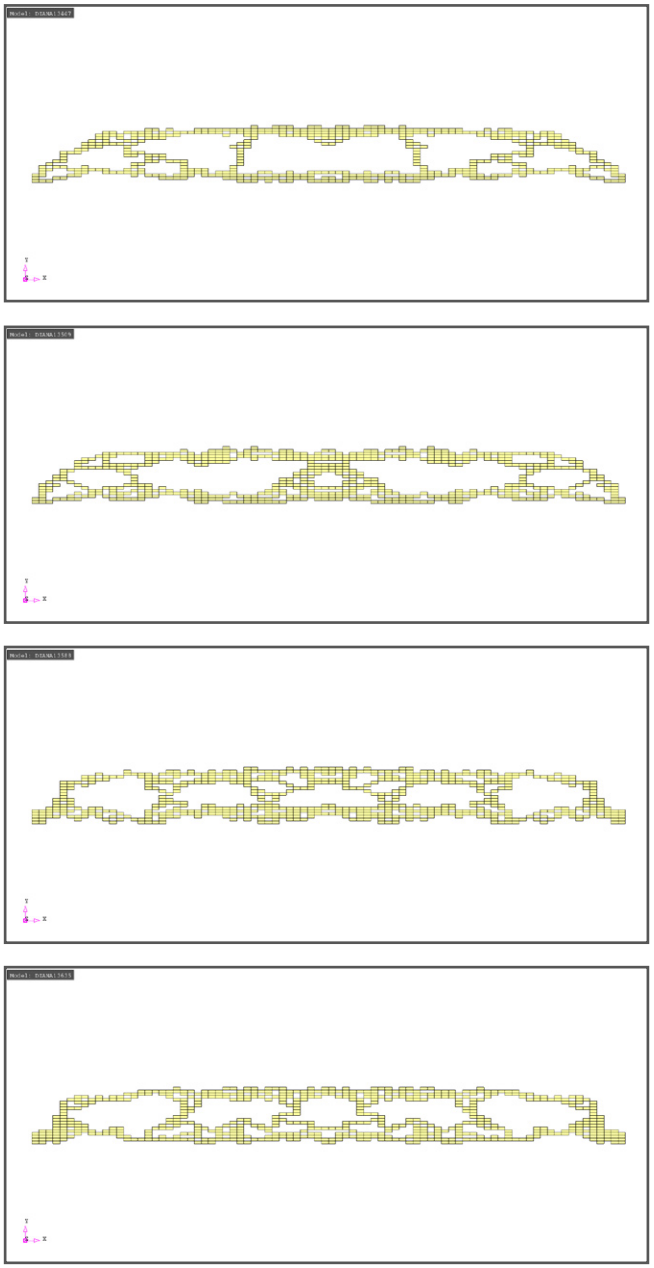
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Rho = 0.40

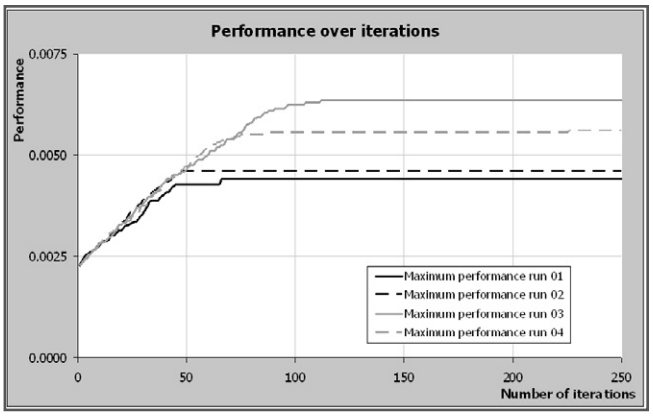


Four runs have been performed resulting in the above four different best solutions.

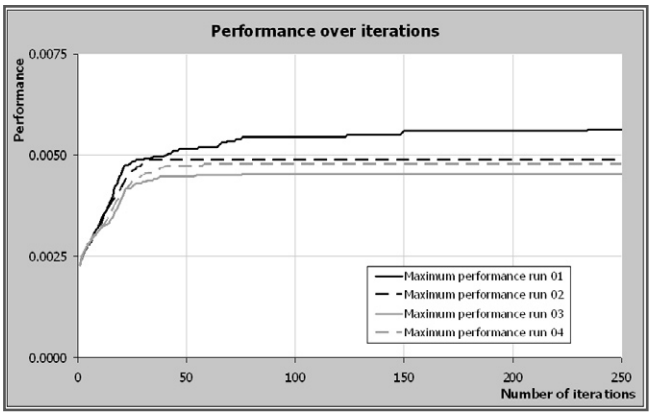
Rho = 0.80



Four runs have been performed resulting in the above four different best solutions.



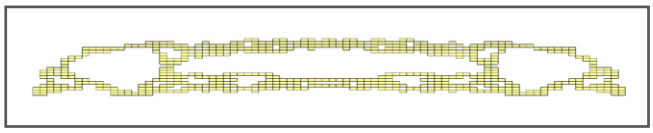
Performance of the four runs that have been performed.



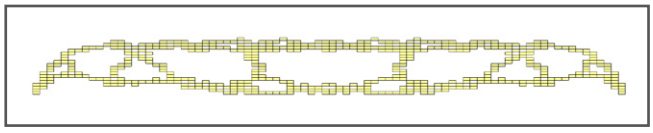
Performance of the four runs that have been performed.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

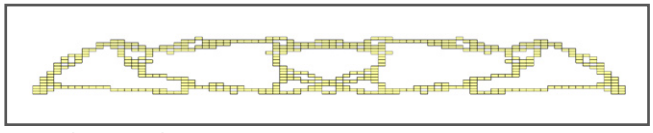
D.2.2 Number of ants



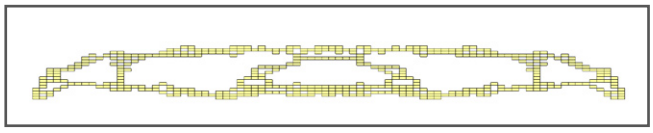
Best solution with 25 ants.



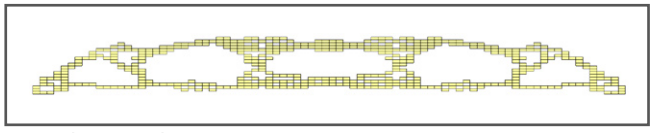
Best solution with 50 ants.



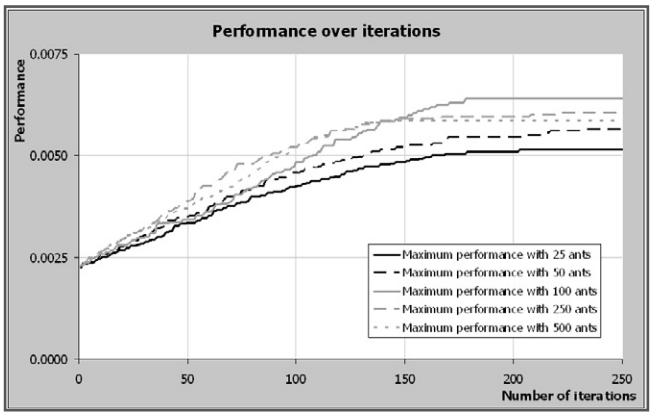
Best solution with 100 ants.



Best solution with 250 ants.

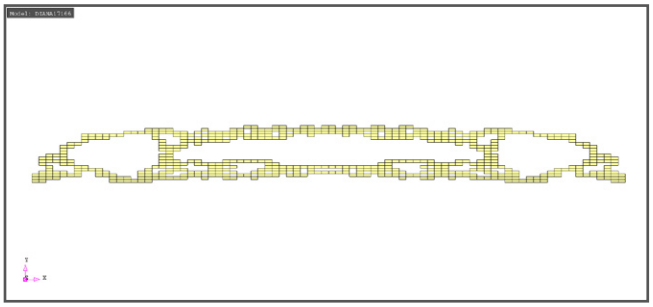


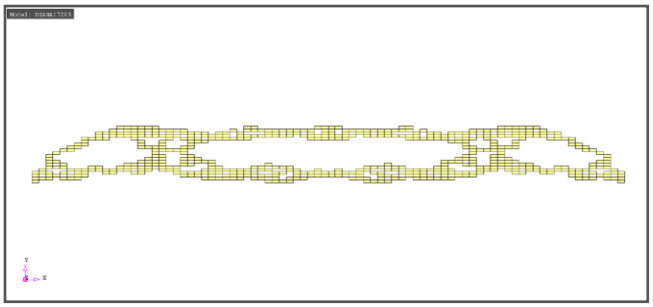
Best solution with 500 ants.

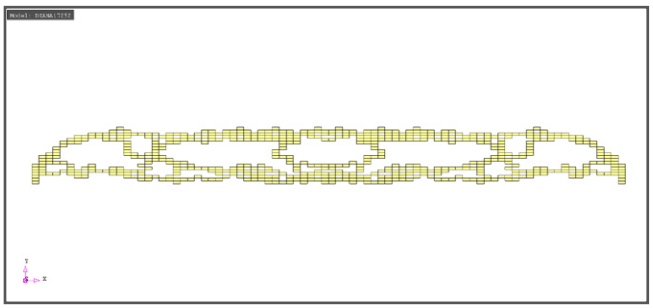


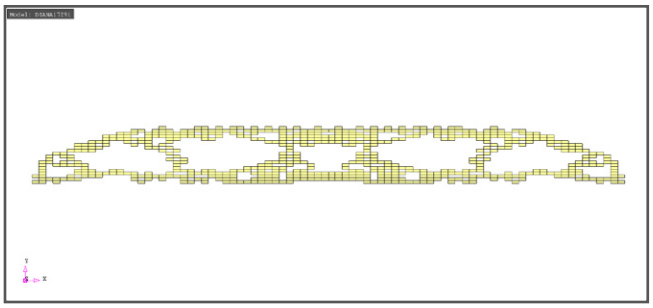
Influence of the number of ants on the performance chart.

25 Ants

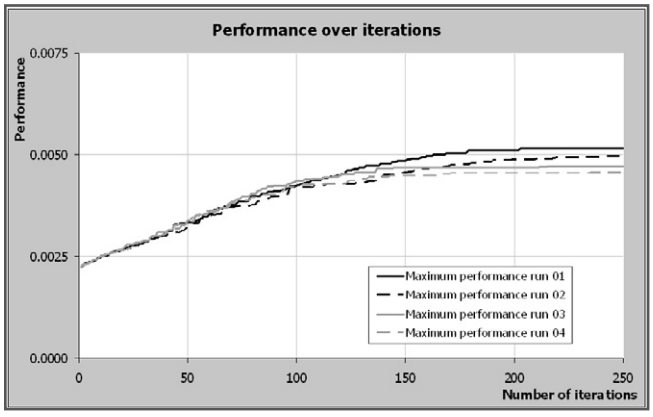








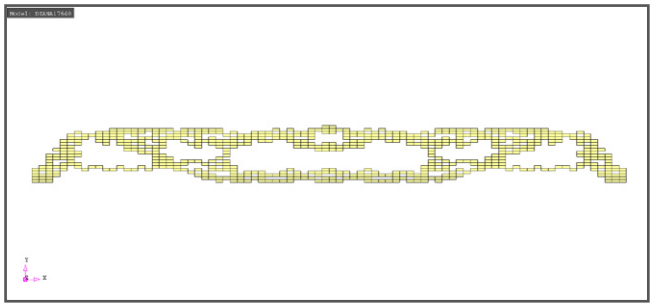
Four runs have been performed resulting in the above four different best solutions.

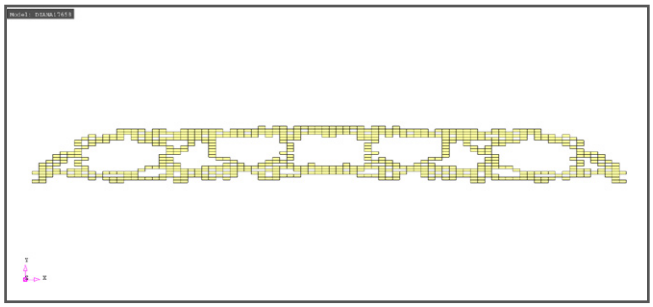


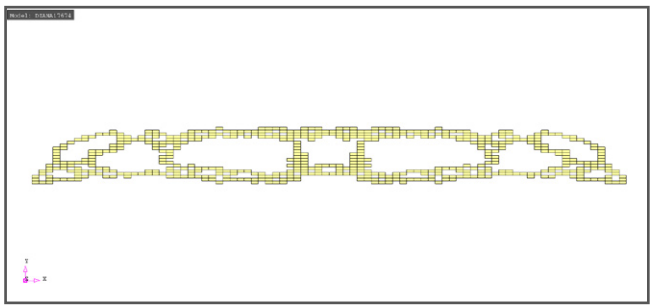
Performance of the four runs that have been performed.

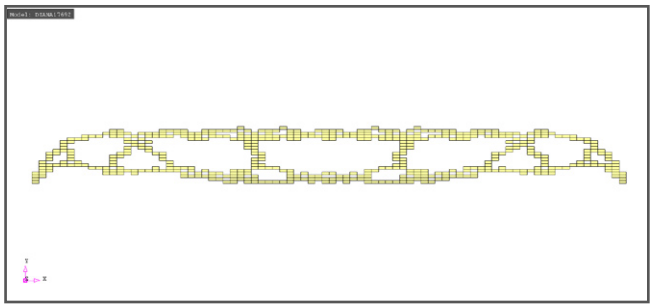
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

50 Ants



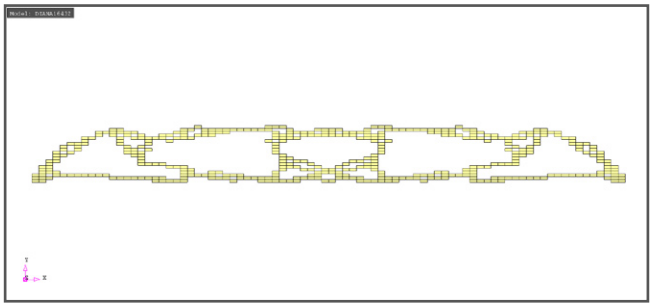


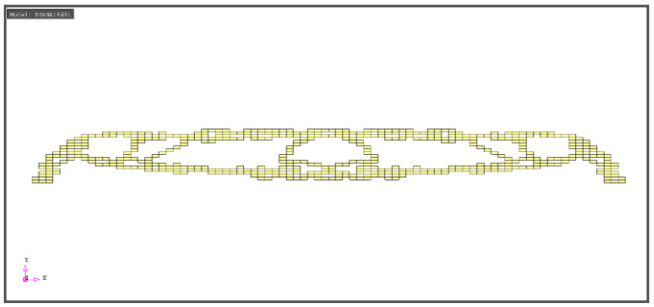


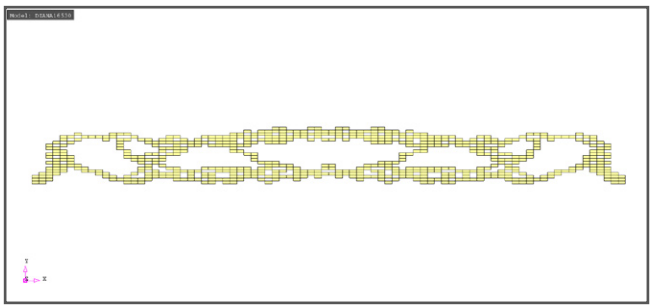


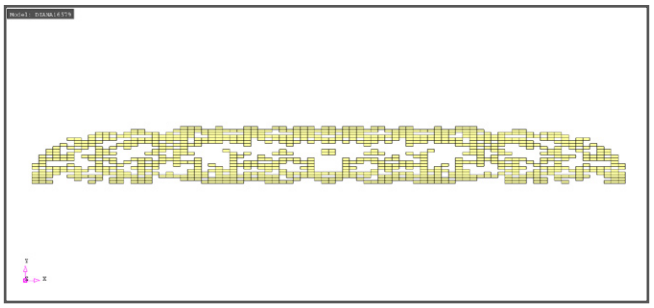
Four runs have been performed resulting in the above four different best solutions.

100 Ants

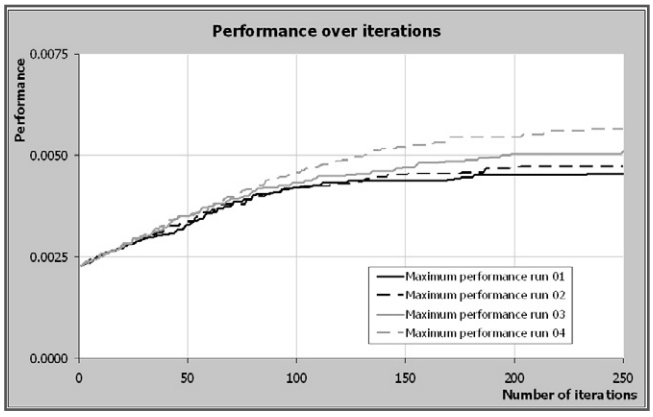




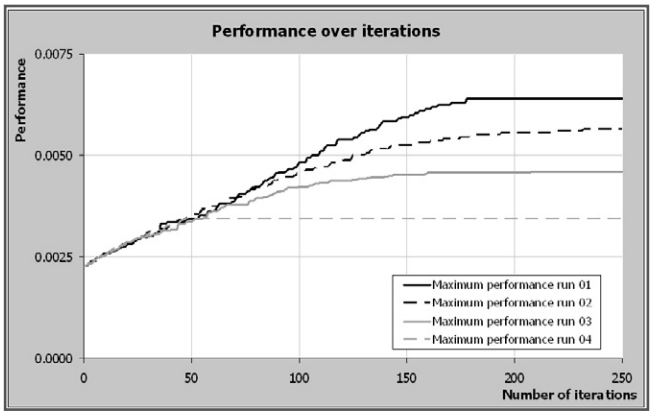




Four runs have been performed resulting in the above four different best solutions.



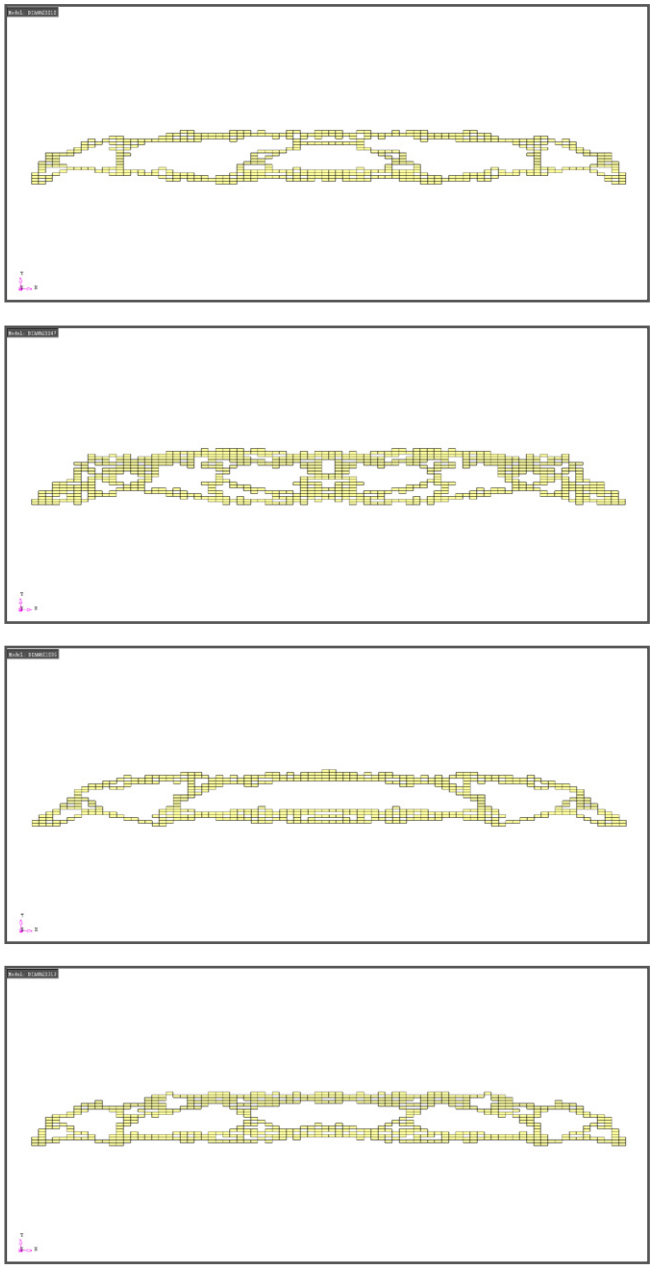
Performance of the four runs that have been performed.



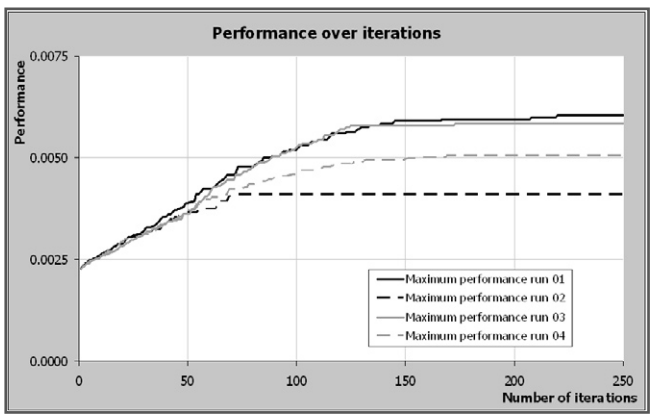
Performance of the four runs that have been performed.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

250 Ants

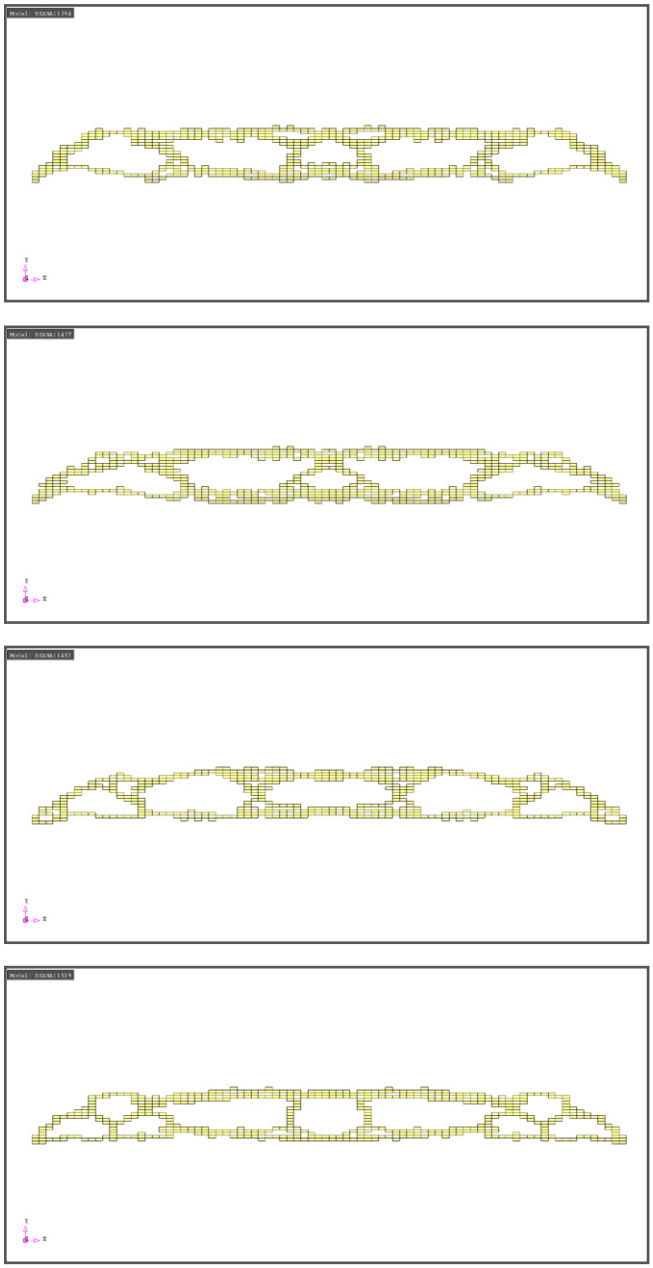


Four runs have been performed resulting in the above four different best solutions.

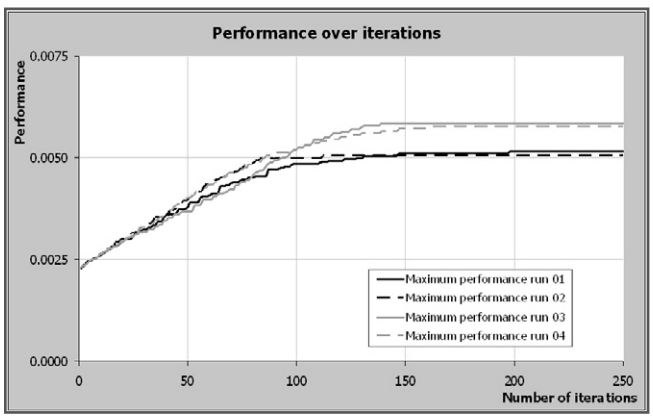


Performance of the four runs that have been performed.

500 Ants



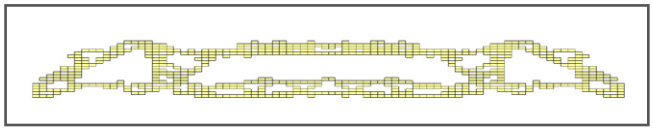
Four runs have been performed resulting in the above four different best solutions.



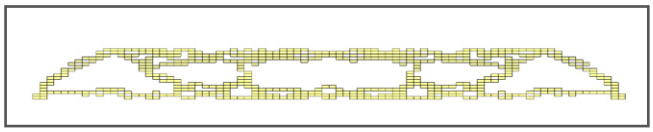
Performance of the four runs that have been performed.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

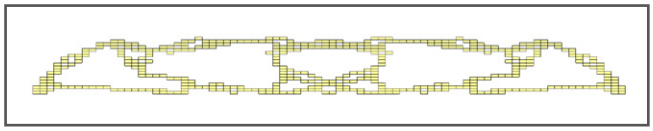
D.2.3 Size of elite



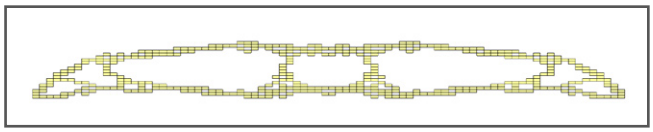
Best solution with 1 elitist ant.



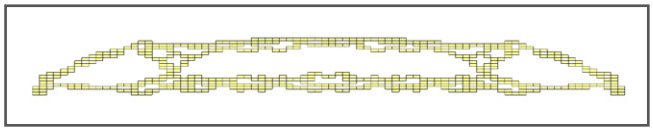
Best solution with 2 elitist ants.



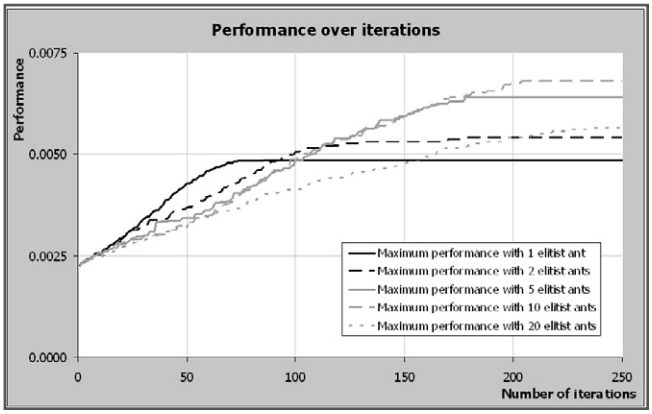
Best solution with 5 elitist ants.



Best solution with 10 elitist ants.

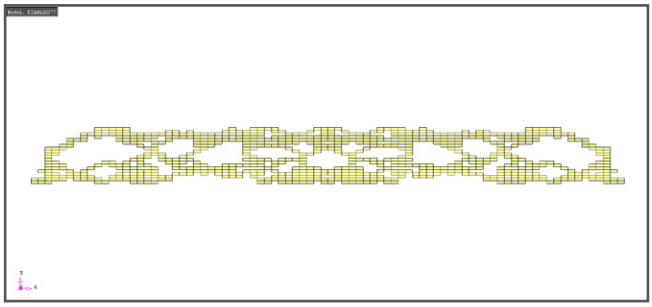


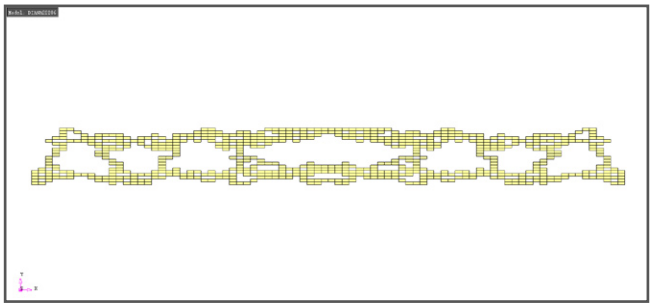
Best solution with 20 elitist ants.

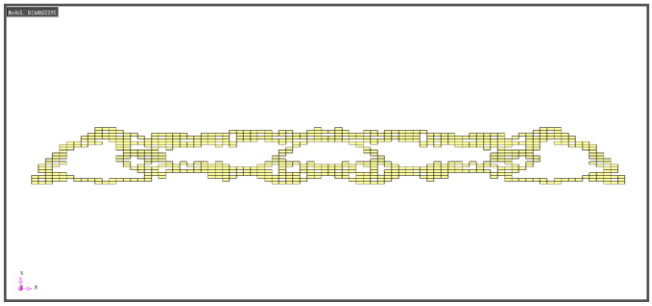


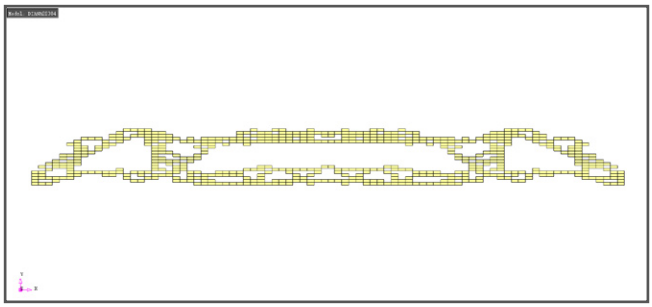
Influence of the number of elitist ants on the performance chart.

1 Elitist ant

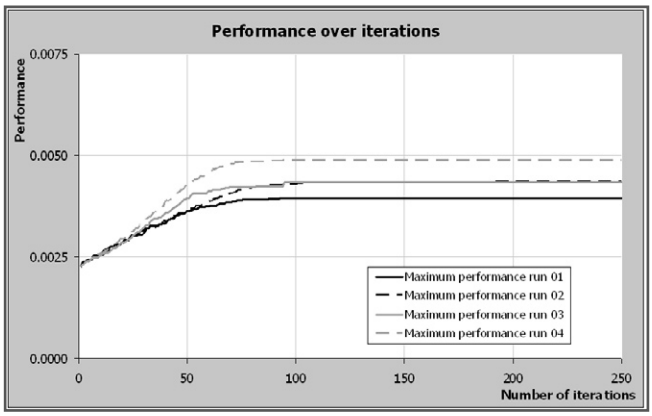








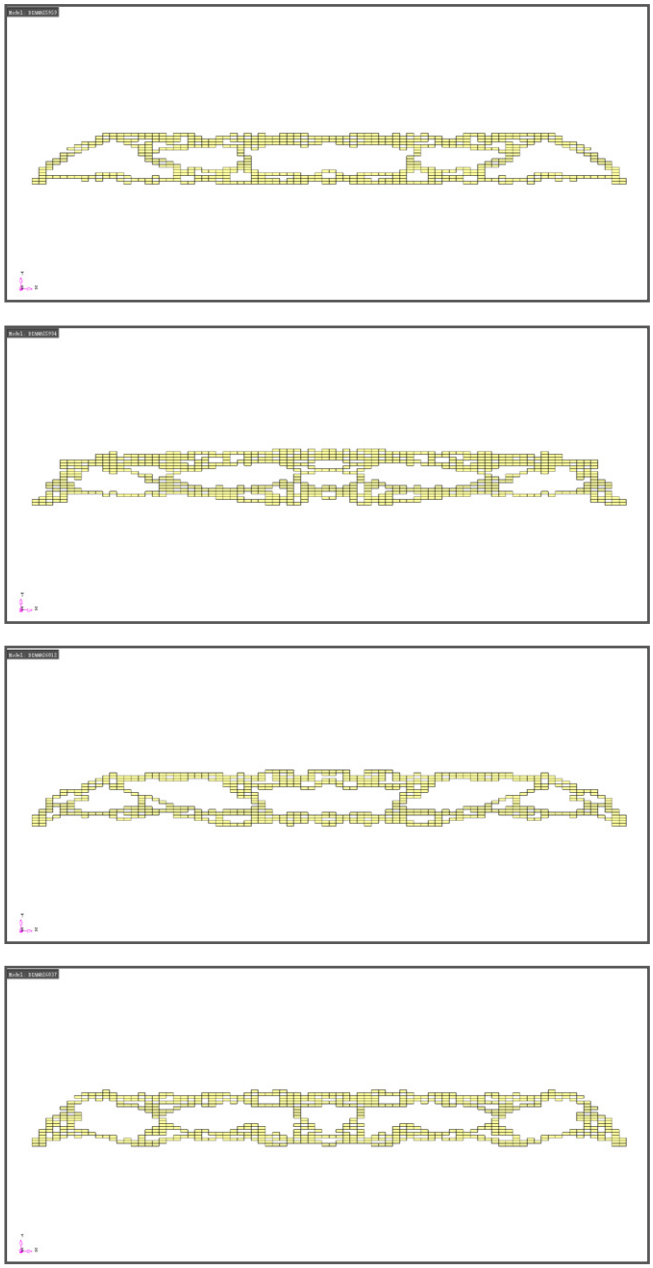
Four runs have been performed resulting in the above four different best solutions.



Performance of the four runs that have been performed.

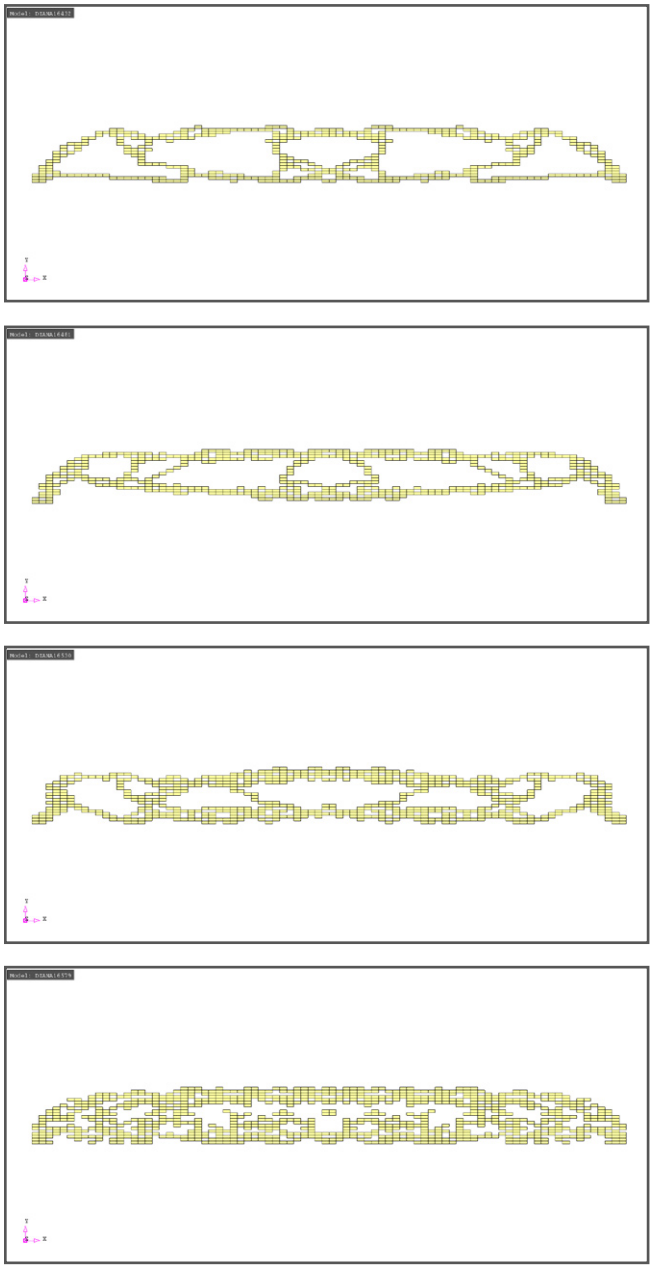
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

2 Elitist ants

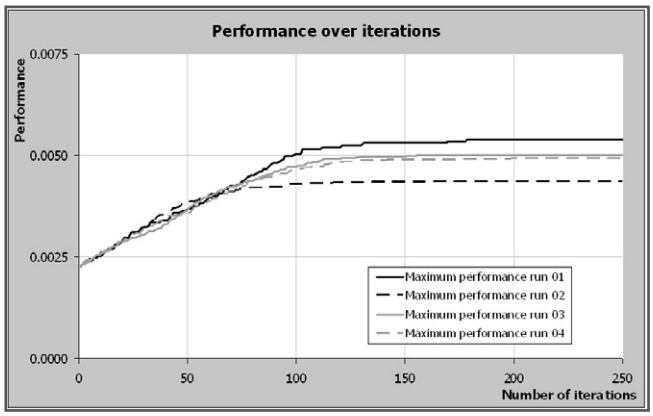


Four runs have been performed resulting in the above four different best solutions.

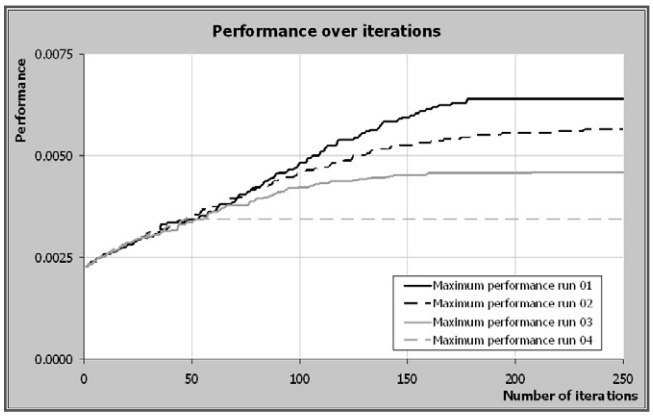
5 Elitist ants



Four runs have been performed resulting in the above four different best solutions.



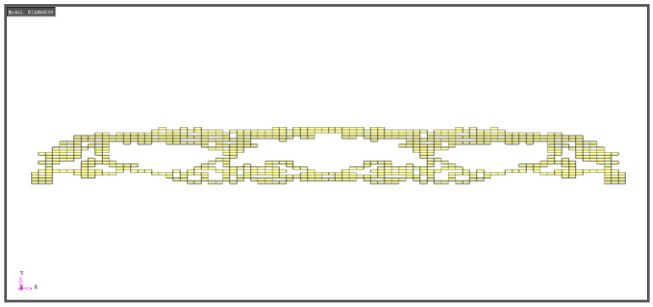
Performance of the four runs that have been performed.

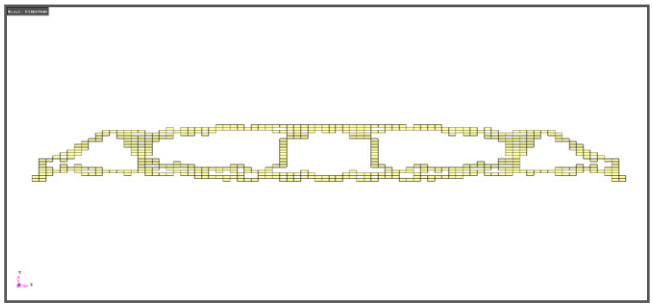


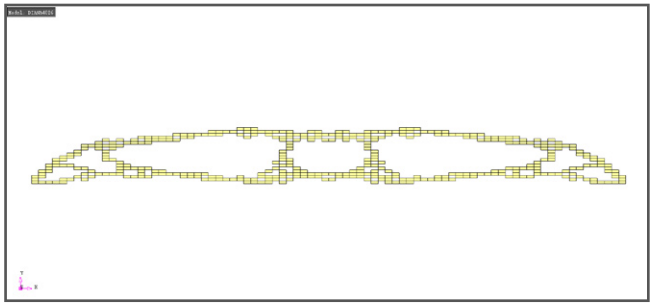
Performance of the four runs that have been performed.

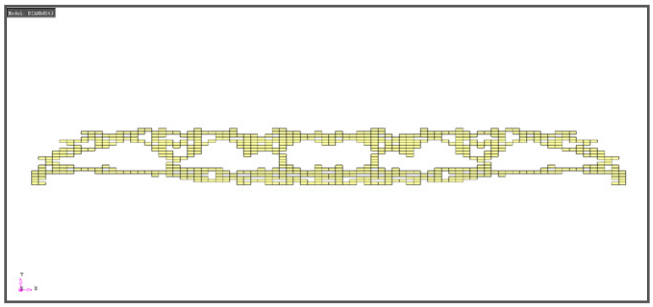
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

10 Elitist ants

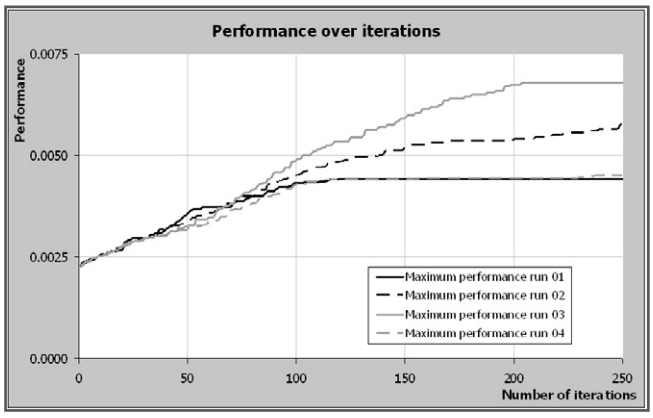






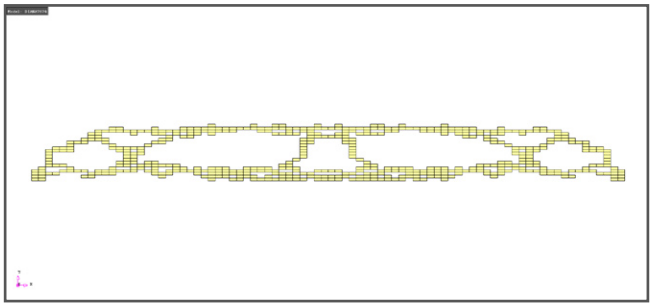


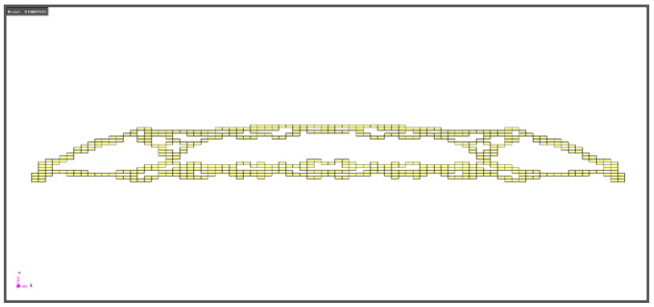
Four runs have been performed resulting in the above four different best solutions.

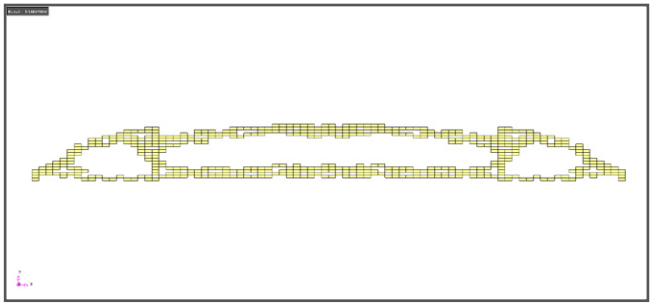


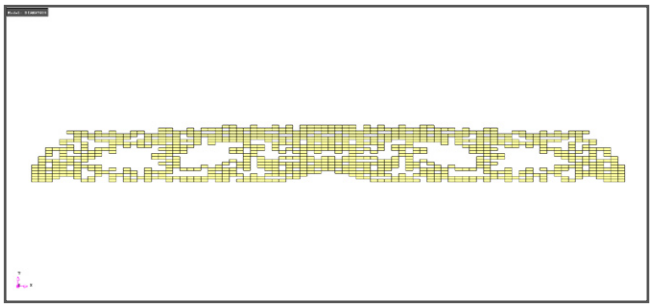
Performance of the four runs that have been performed.

20 Elitist ants

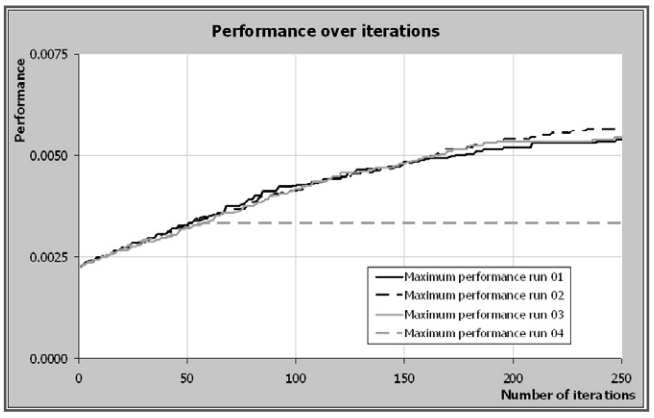








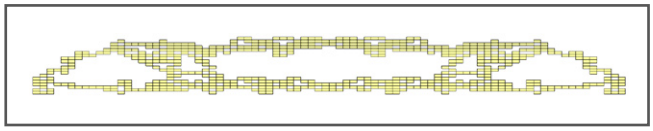
Four runs have been performed resulting in the above four different best solutions.



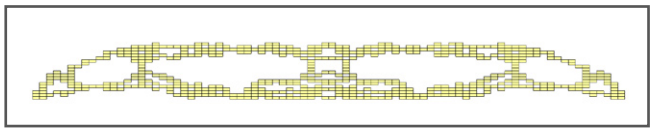
Performance of the four runs that have been performed.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

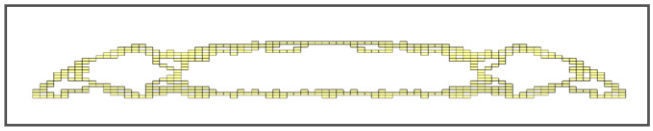
D.2.4 Stress daemon



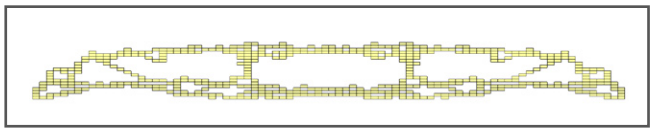
Best solution with one adjustment to 50% of the performance.



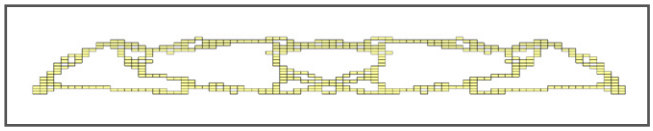
Best solution with two adjustments to 50% and 70% of the performance.



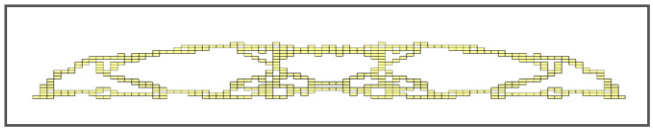
Best solution with two adjustments to 50% and 90% of the performance.



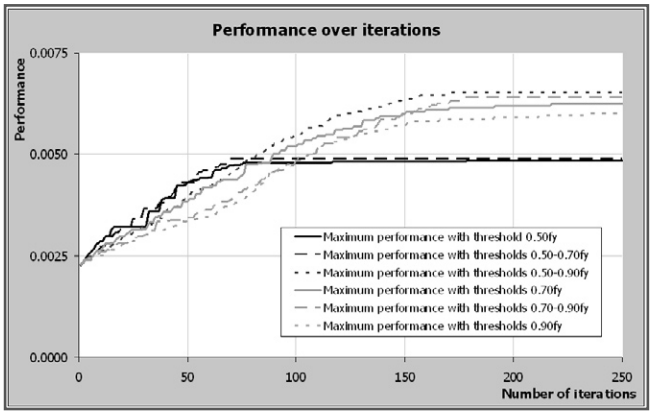
Best solution with one adjustment to 70% of the performance.



Best solution with two adjustments to 70% and 90% of the performance.

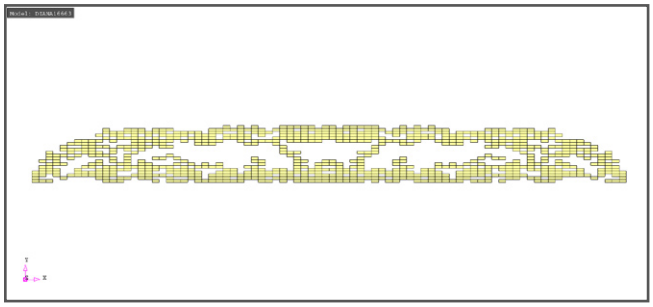


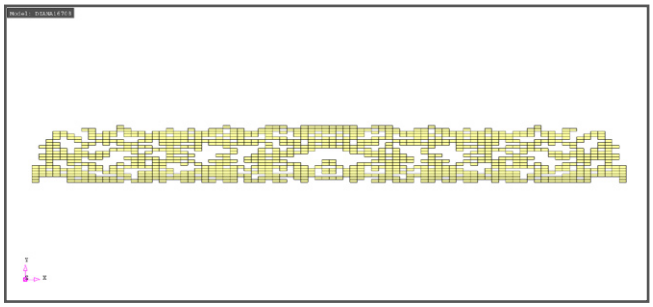
Best solution with one adjustment to 90% of the performance.

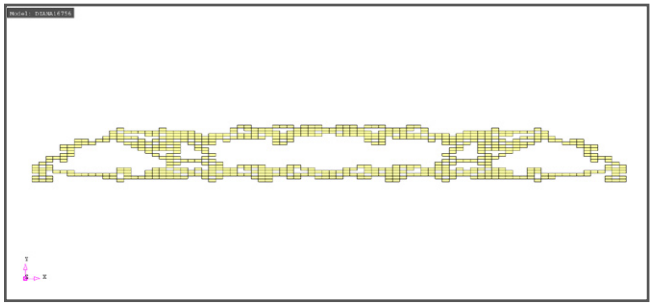


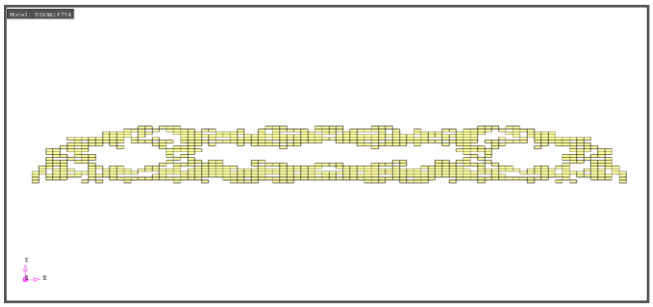
Performance charts of the runs with different stress daemons.

Adjustment to 50%

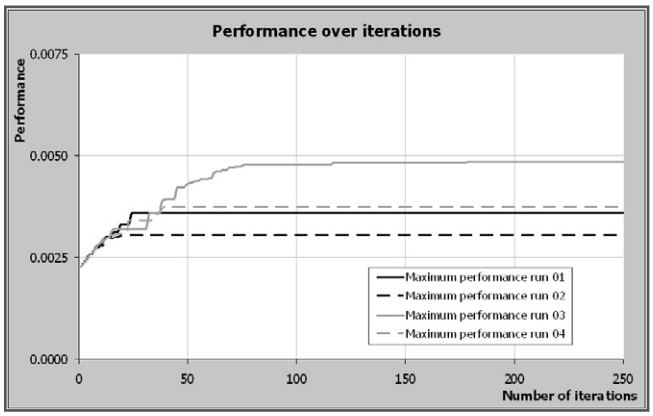








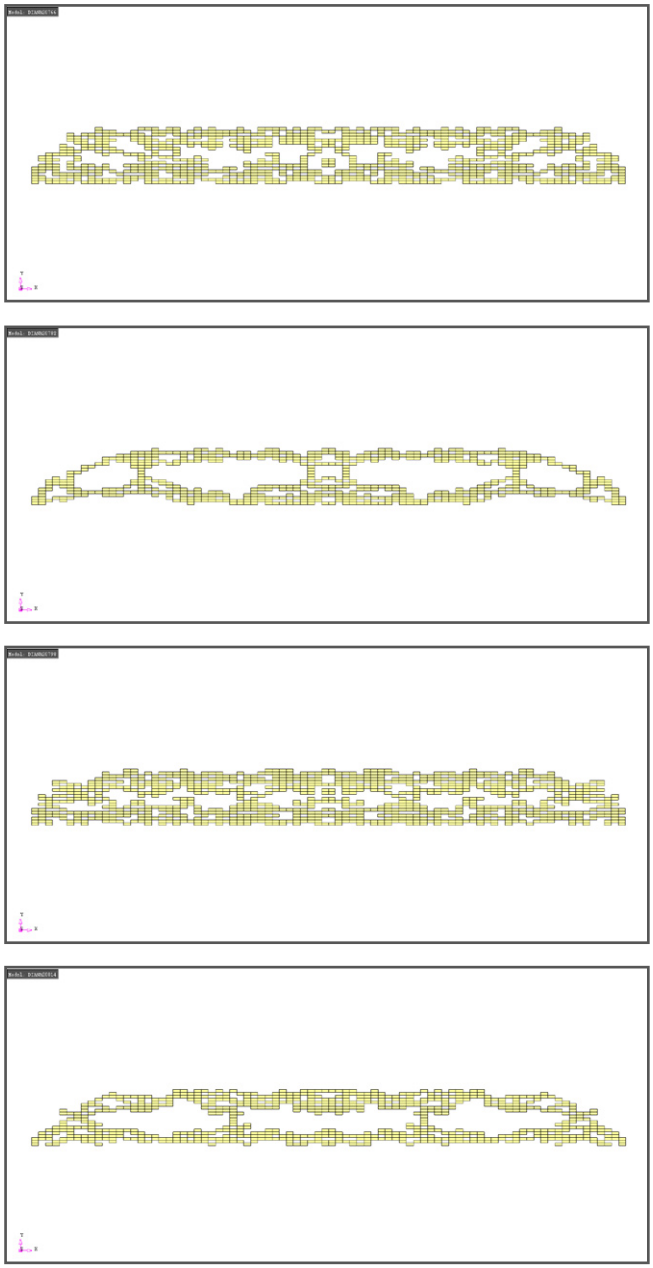
Four runs have been performed resulting in the above four different best solutions.



Performance of the four runs that have been performed.

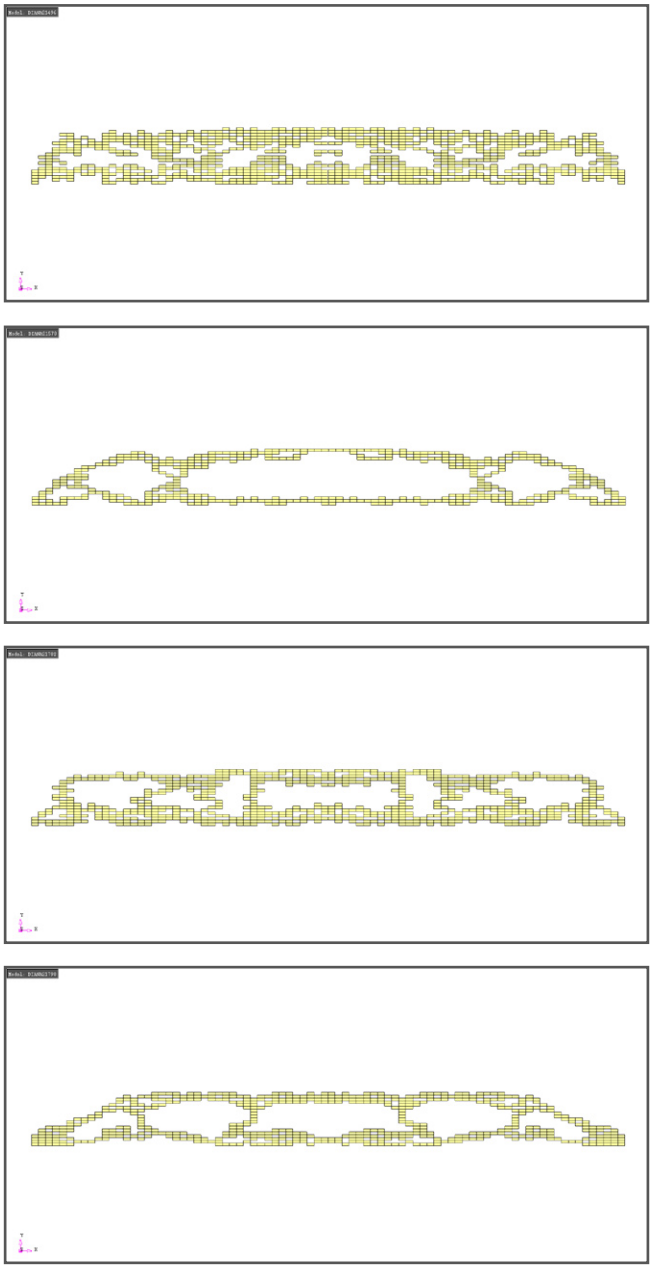
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Adjustment to 50%-70%

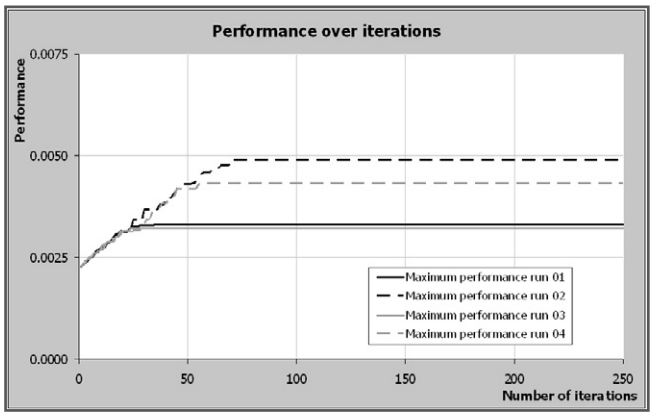


Four runs have been performed resulting in the above four different best solutions.

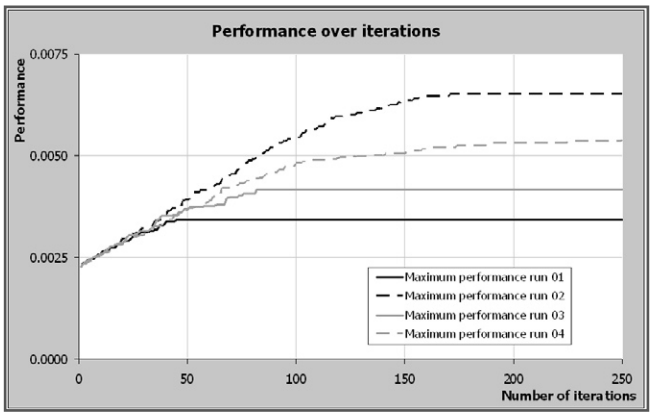
Adjustment to 50%-90%



Four runs have been performed resulting in the above four different best solutions.



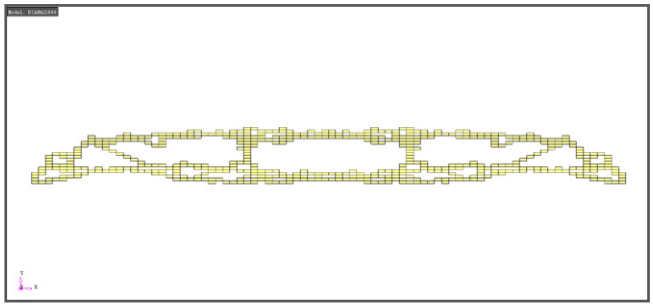
Performance of the four runs that have been performed.

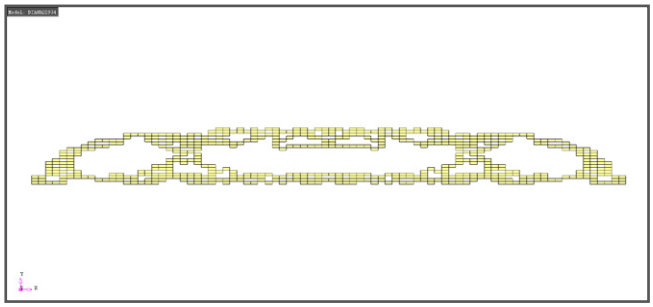


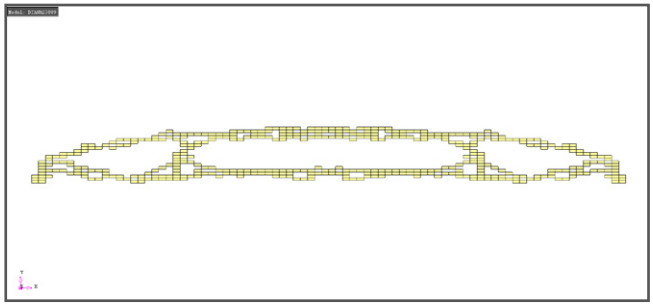
Performance of the four runs that have been performed.

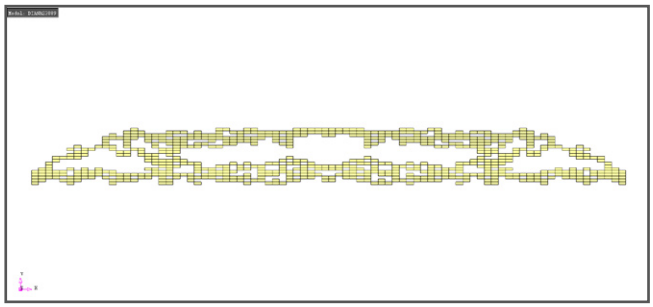
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Adjustment to 70%



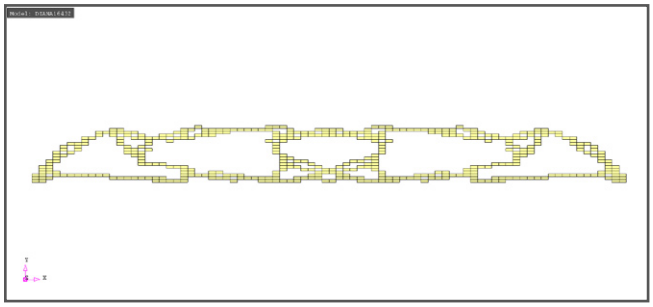


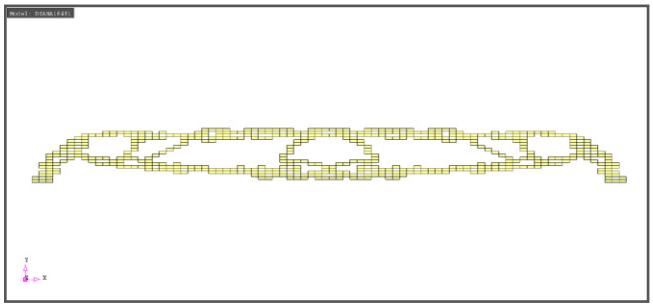


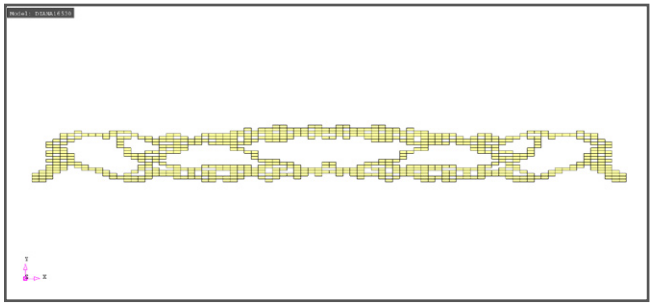


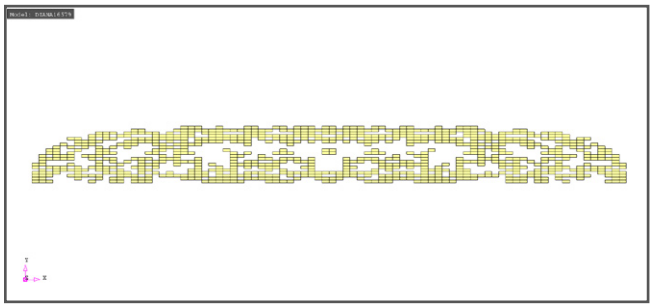
Four runs have been performed resulting in the above four different best solutions.

Adjustment to 70%-90%

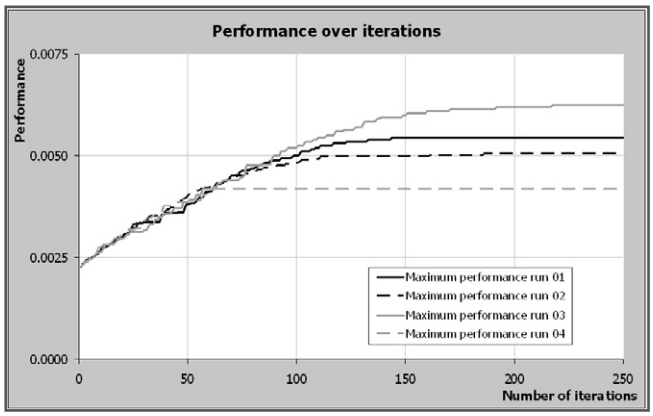




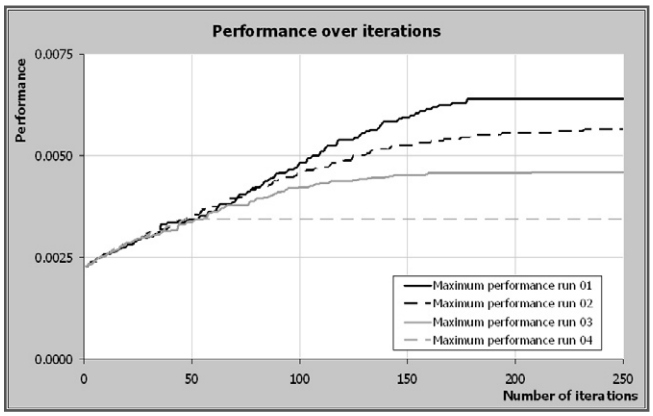




Four runs have been performed resulting in the above four different best solutions.



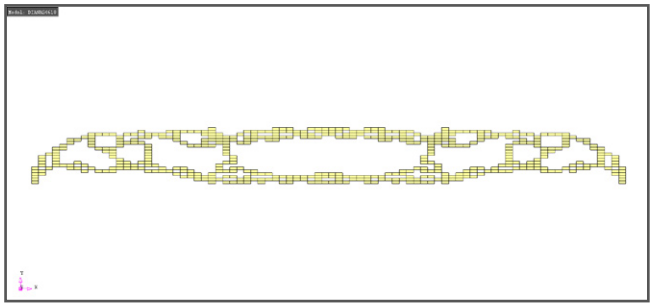
Performance of the four runs that have been performed.

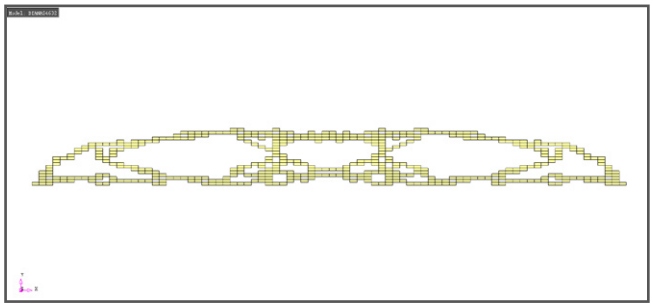


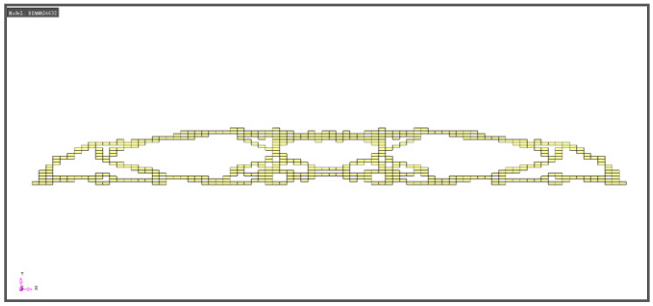
Performance of the four runs that have been performed.

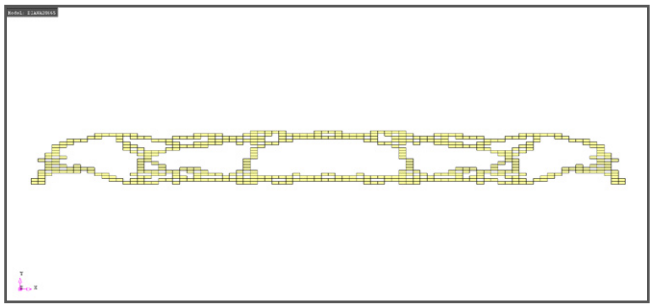
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Adjustment to 90%

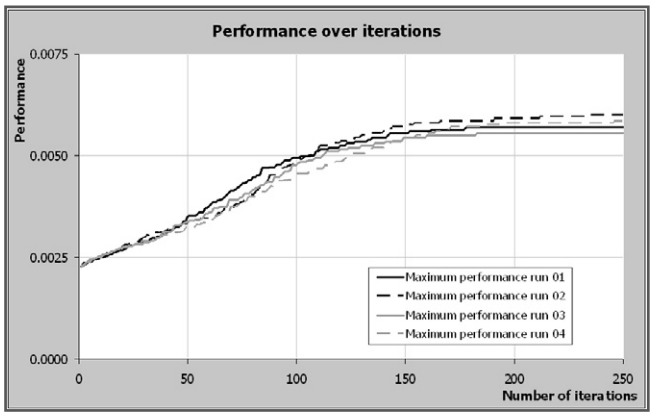








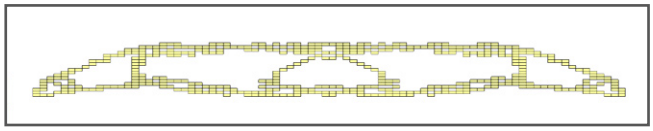
Four runs have been performed resulting in the above four different best solutions.



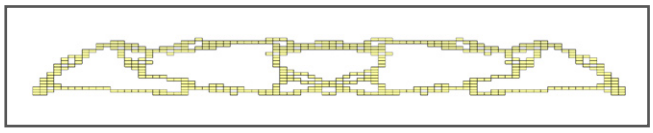
Performance of the four runs that have been performed.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

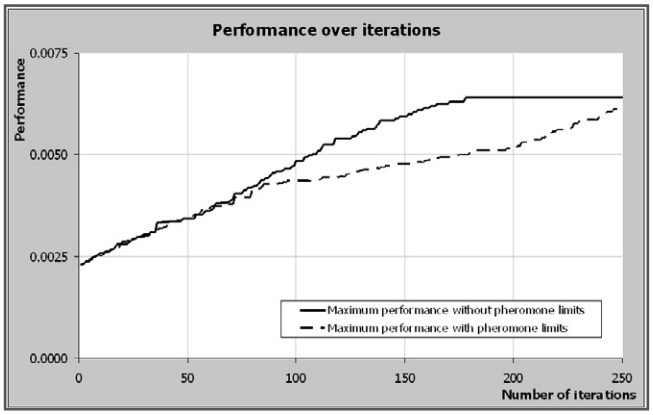
D.2.5 Pheromone limits



Best solution of the runs with pheromone limits.

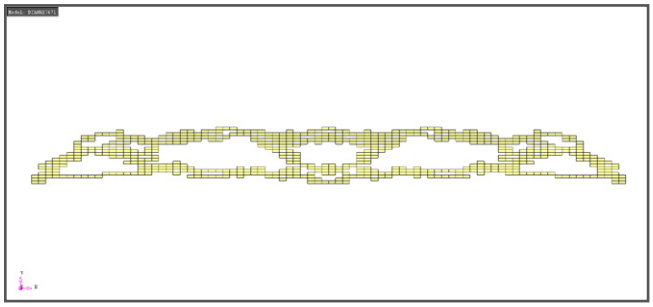


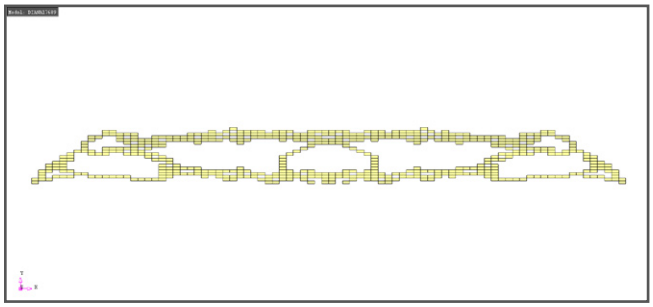
Best solution of the runs without pheromone limits.

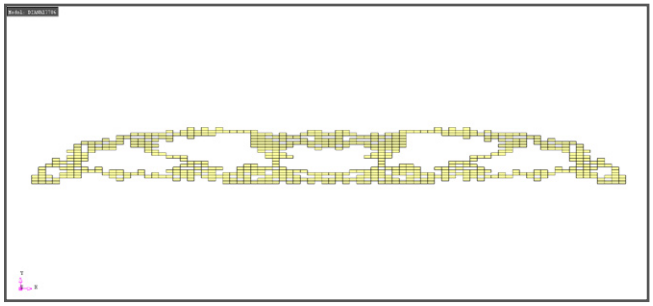


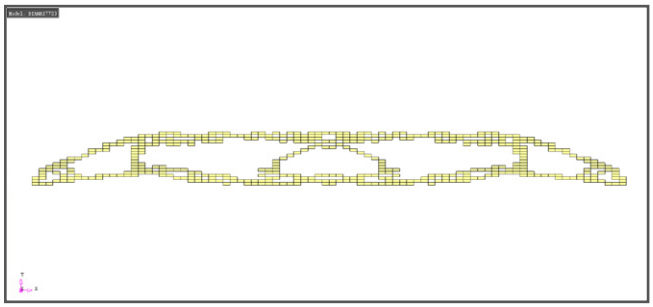
Performance charts of the best runs with and without pheromone limits.

Limits 0.05-0.99

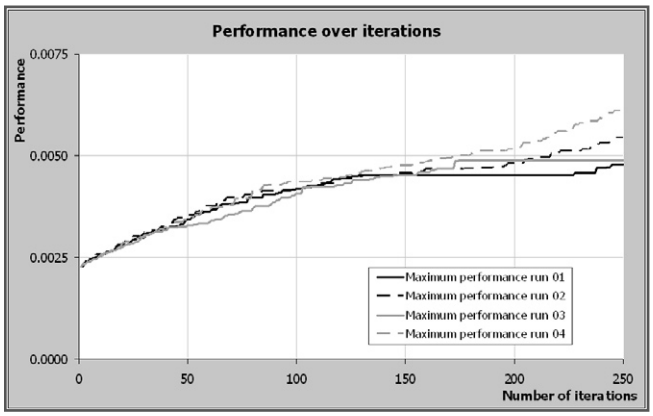








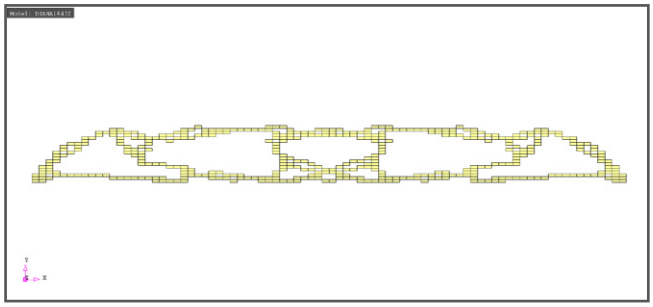
Four runs have been performed resulting in the above four different best solutions.

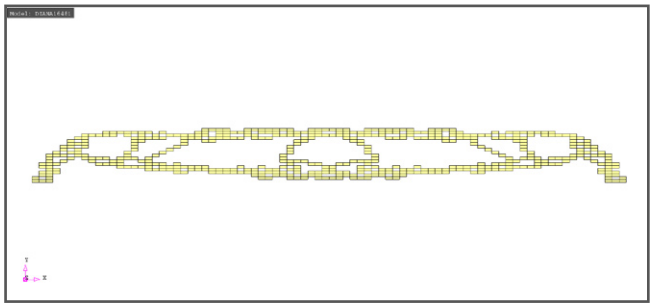


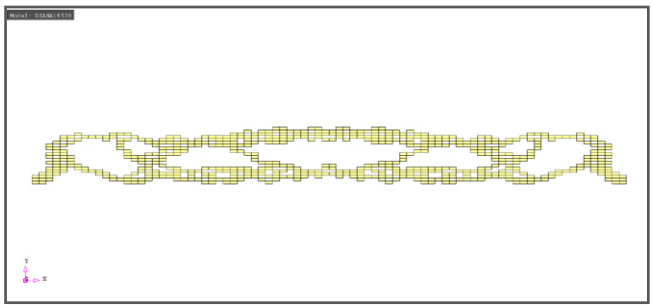
Performance of the four runs that have been performed.

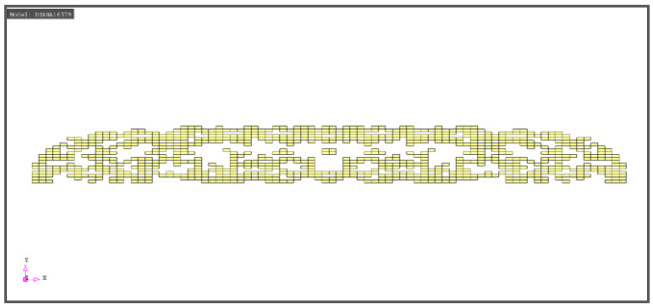
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Limits 0.00-1.00

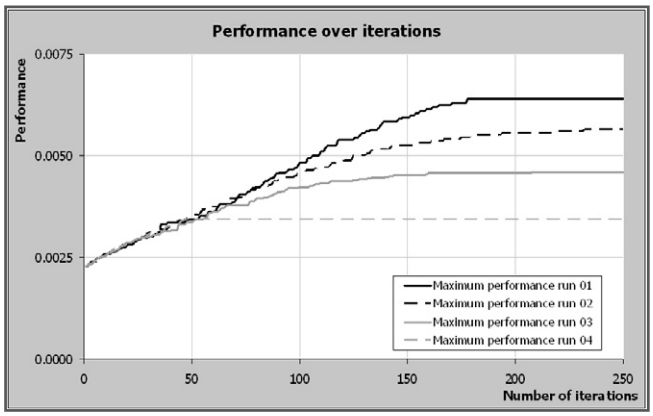








Four runs have been performed resulting in the above four different best solutions.



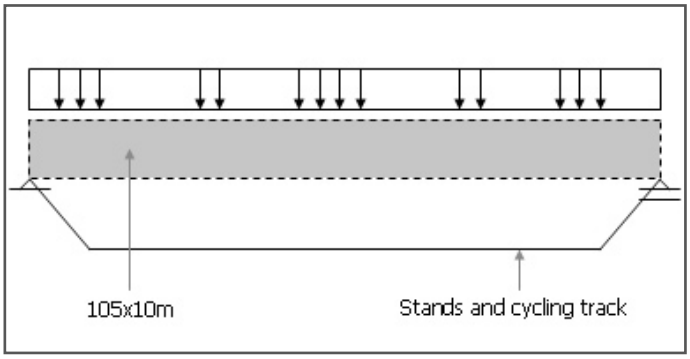
Performance of the four runs that have been performed.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

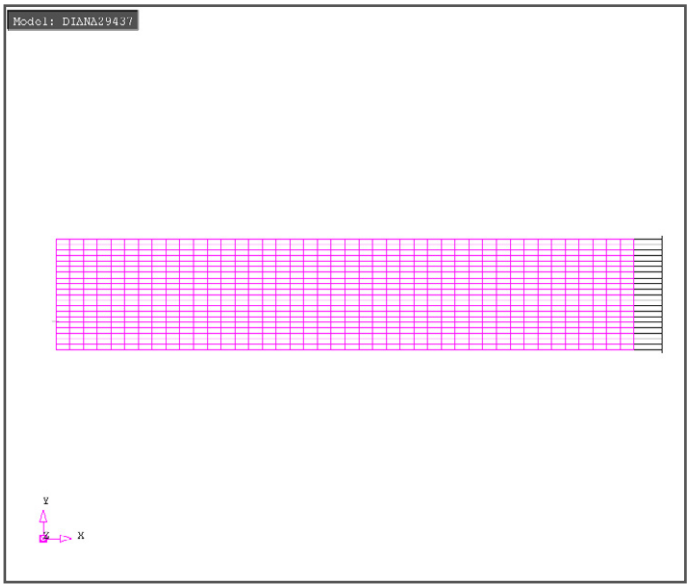
D.3 Performance sensitivity

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	250	
Ants	100	
Elitist ants	5	
Ant runs	25000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.00E+07	Pa
Fy;min	-8.00E+07	Pa
Ft	5.00E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.7	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.24	kN/m²
Load case ULS2	0.98	kN/m²
Load case SLS1	2.56	kN/m²
Load case SLS2	1.05	kN/m²
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	1:47:30	hh:mm:ss
Run time (average)	6450	s
Average time per ant run	0.258	s

Job information.



Design space and boundary conditions.



Mesh.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ (\sigma_{i;max} - \sigma_{i;min}) > (\sigma_{allowable;max} - f_t) \\ \dfrac{1}{C_V V + C_S S + C_P P V_p + C_H H} & else \end{cases}$$

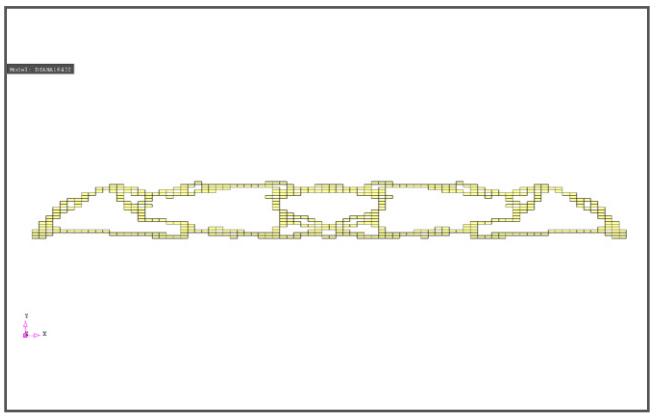
Performance function.

	Set 1	Set 2	Set 3
C_V	2.00	2.00	2.00
C_S	0.30	0.05	0.50
C_P	0.50	20.0	10.0
C_H	20.0	20.0	20.0

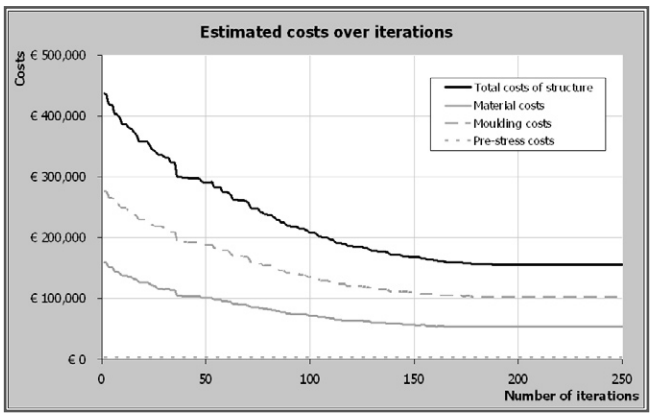
Parameter sets.

The performance function consists of four parameters, that are all multiplied with a constant to estimate the cost of the structure. The inverse of this cost is the performance. In this appendix the estimated costs are plotted as well as the estimated costs of each parameters that takes part in the performance function. This way insight is generated in the influence of the parameters in the performance function.

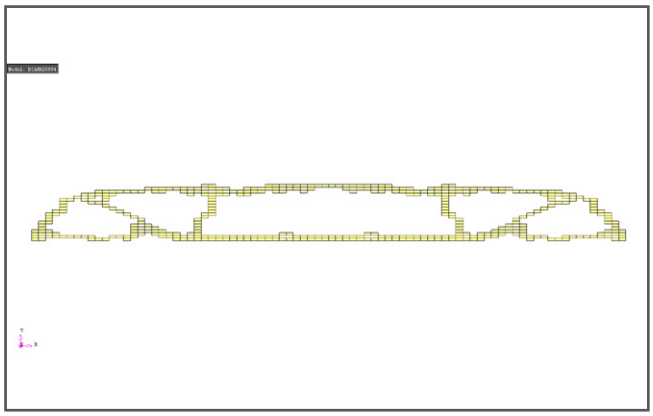
Three sets of constants are tested. Any visual output is mirrored along the line of symmetry; only half the problem is analysed.



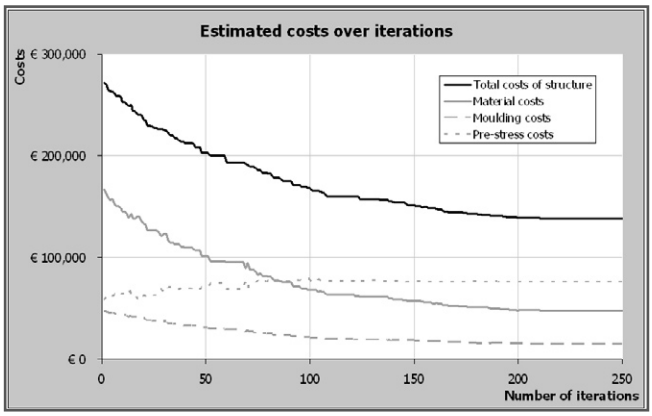
Best structure found with the performance function defined by set 1.



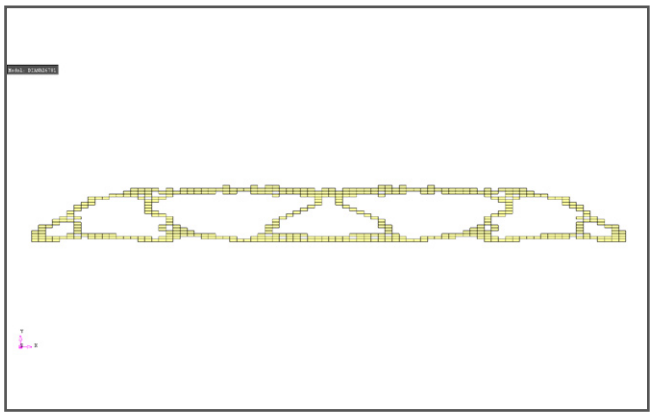
Estimated costs of the best structure found in four runs with the performance function defined by set 1.



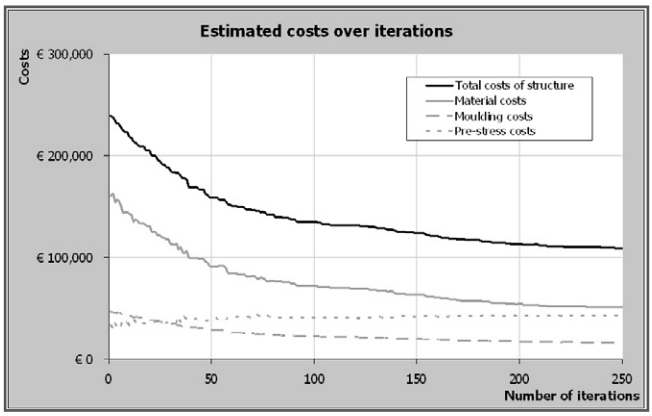
Best structure found with the performance function defined by set 2.



Estimated costs of the best structure found in four runs with the performance function defined by set 2.

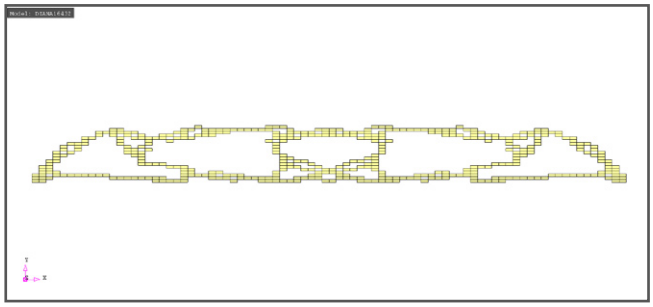


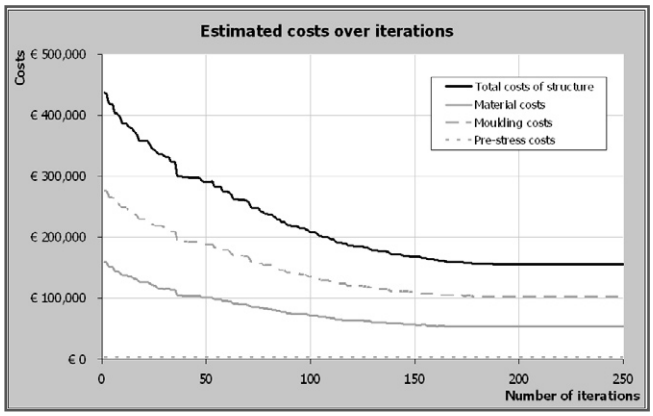
Best structure found with the performance function defined by set 3.



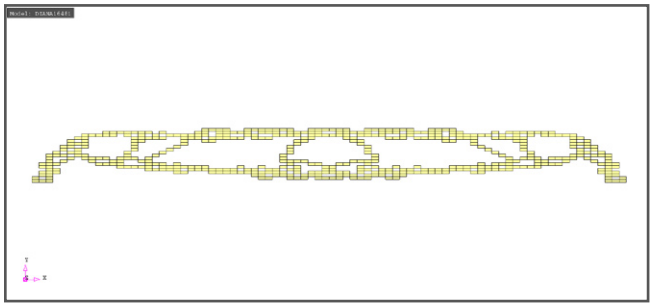
Estimated costs of the best structure found in four runs with the performance function defined by set 3.

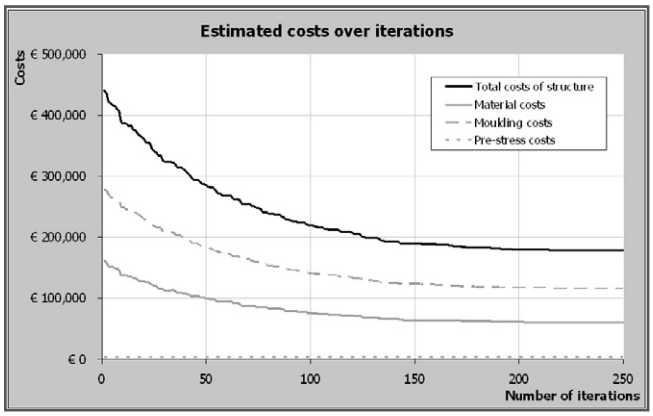
Visual output is mirrored along the line of symmetry; only half the problem is analysed.



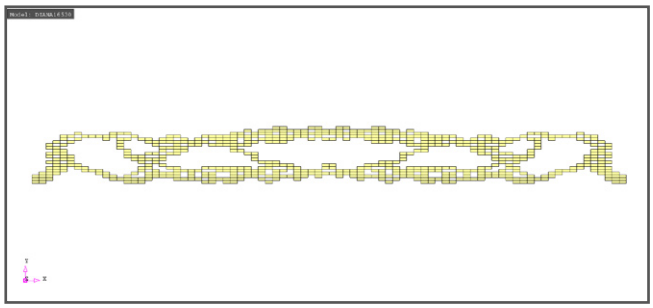


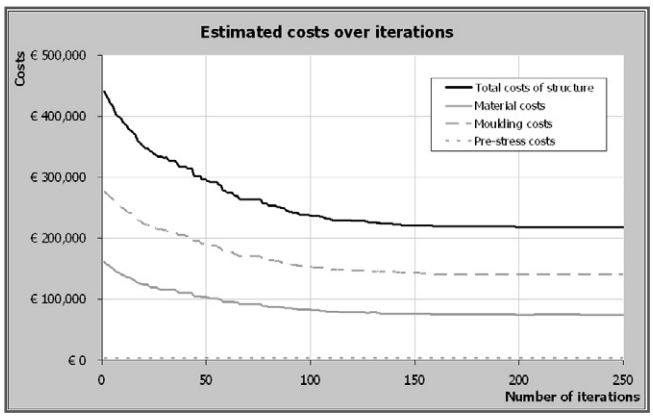
Run 1: Best solution and cost estimation over iterations.



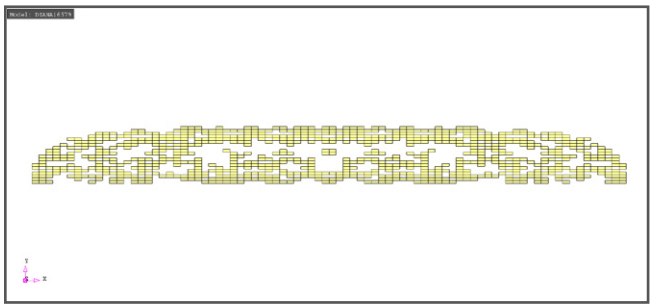


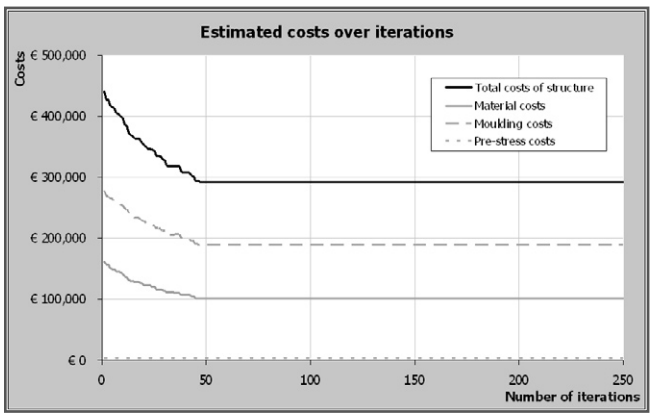
Run 2: Best solution and cost estimation over iterations.





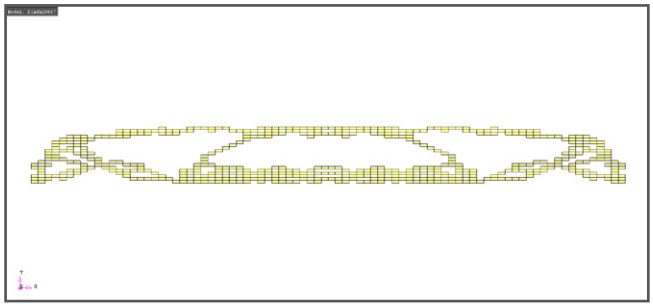
Run 3: Best solution and cost estimation over iterations.

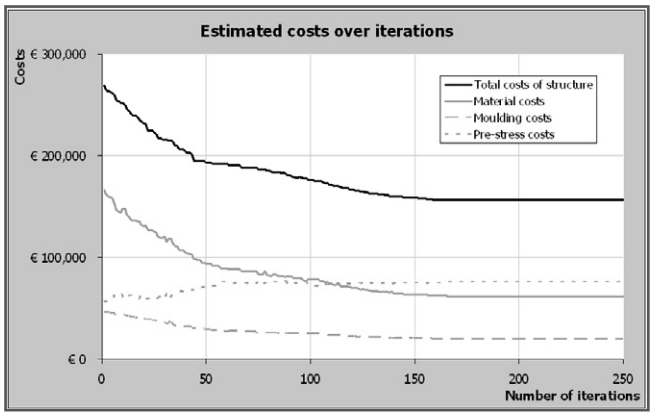




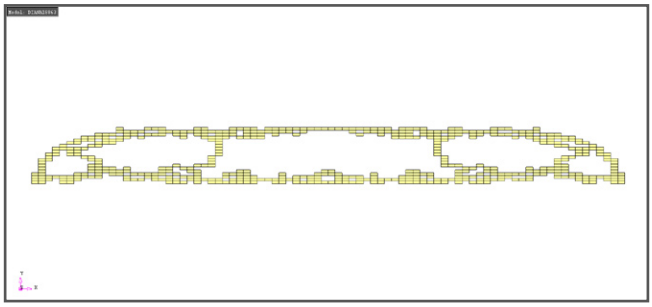
Run 4: Best solution and cost estimation over iterations.

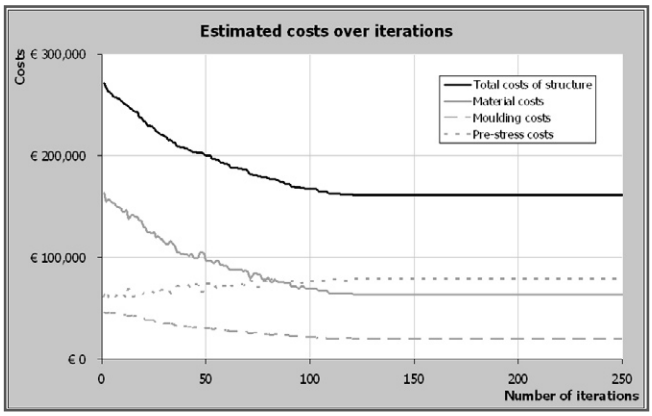
Visual output is mirrored along the line of symmetry; only half the problem is analysed.



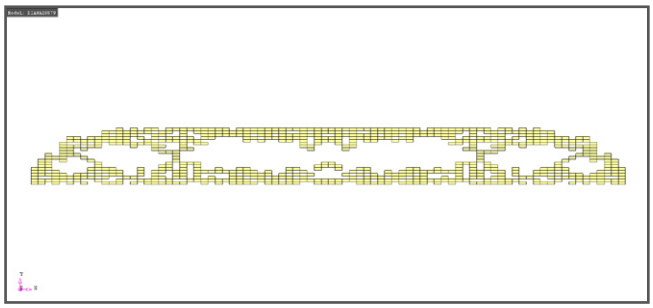


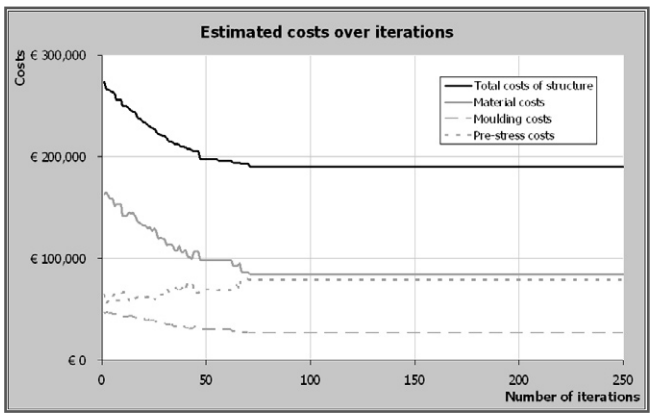
Run 1: Best solution and cost estimation over iterations.



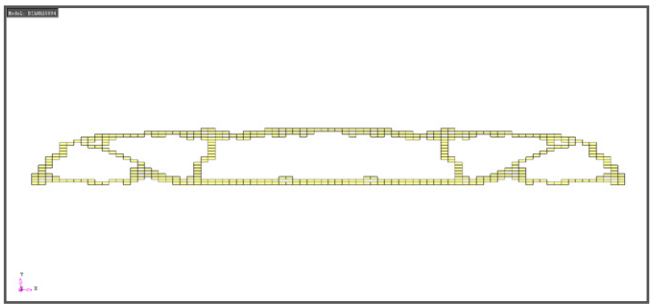


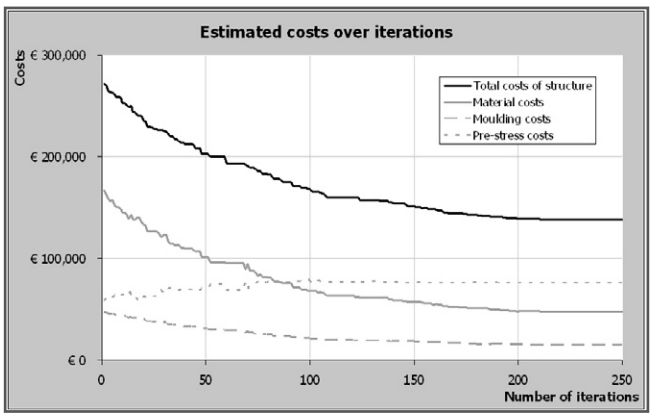
Run 2: Best solution and cost estimation over iterations.





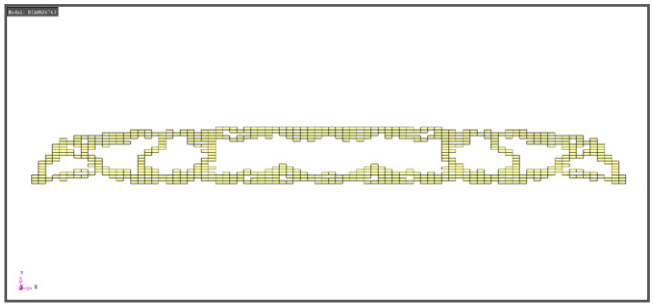
Run 3: Best solution and cost estimation over iterations.

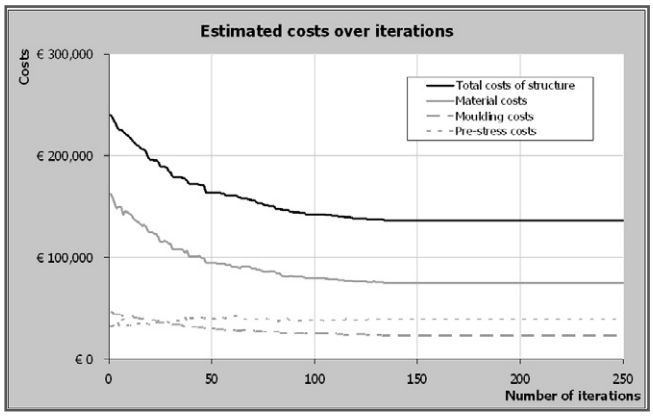




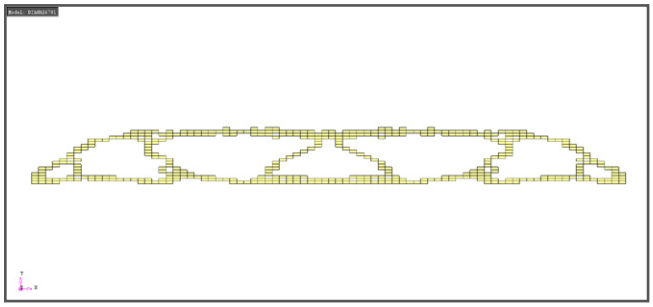
Run 4: Best solution and cost estimation over iterations.

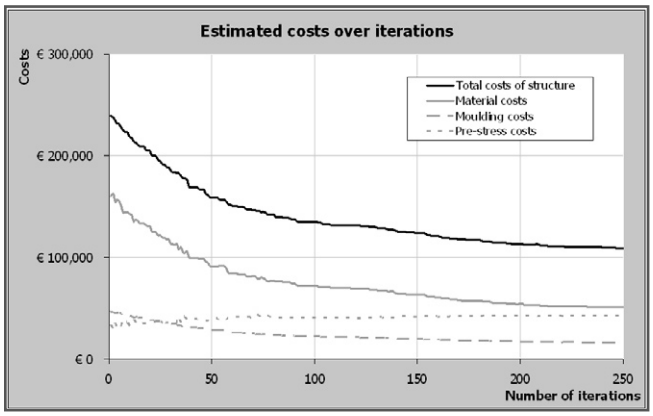
Visual output is mirrored along the line of symmetry; only half the problem is analysed.



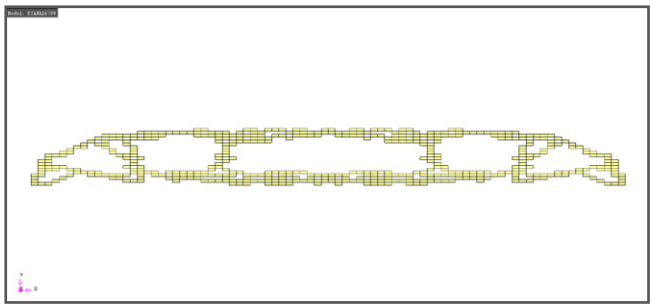


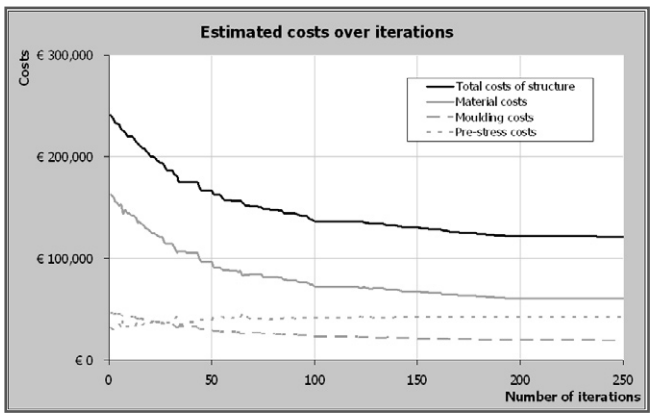
Run 1: Best solution and cost estimation over iterations.



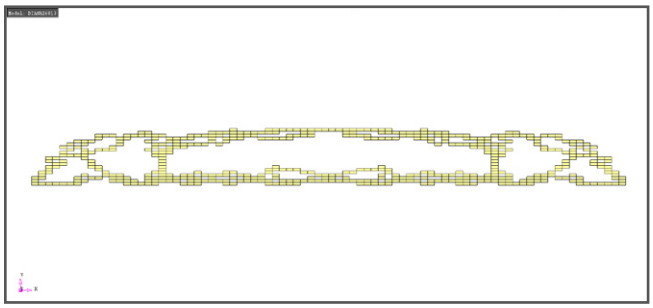


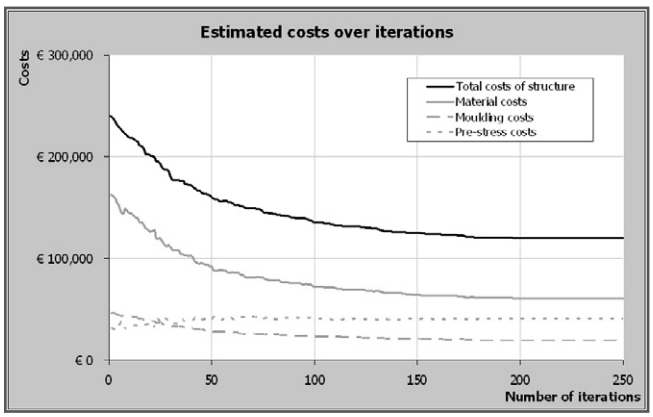
Run 2: Best solution and cost estimation over iterations.





Run 3: Best solution and cost estimation over iterations.





Run 4: Best solution and cost estimation over iterations.

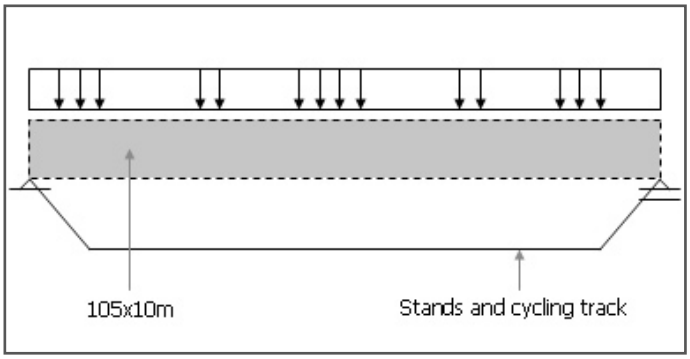
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

D.4 Case sensitivity

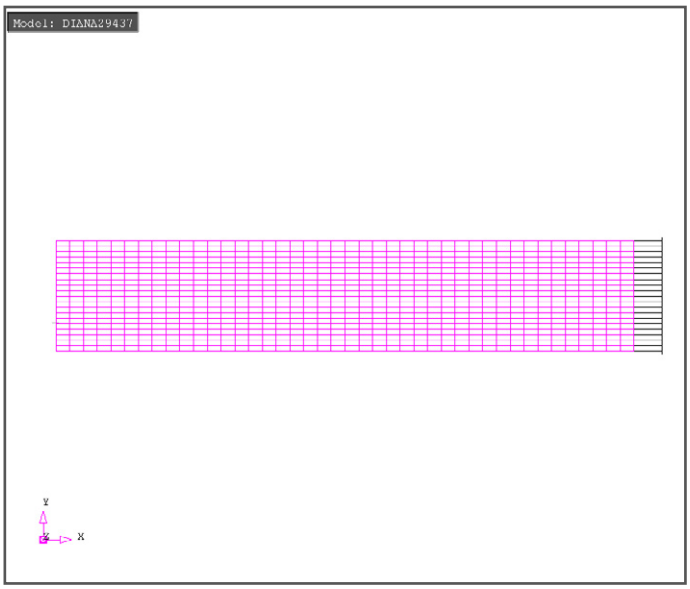
D.4.1 Coarse model

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	500	
Ants	100	
Elitist ants	8	
Ant runs	50000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m²
Load case ULS2	0.20	kN/m²
Load case SLS1	2.26	kN/m²
Load case SLS2	0.85	kN/m²
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	4:41:07	hh:mm:ss
Run time (average)	16867	s
Average time per ant run	0.337	s

Job information.



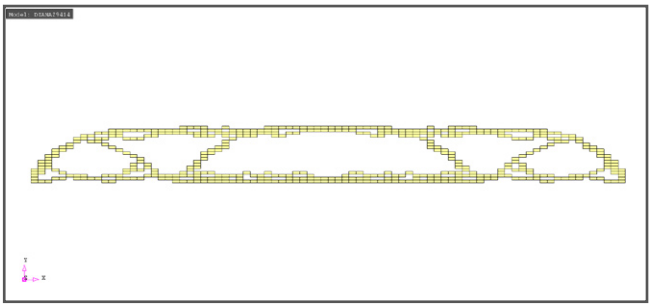
Design space and boundary conditions.



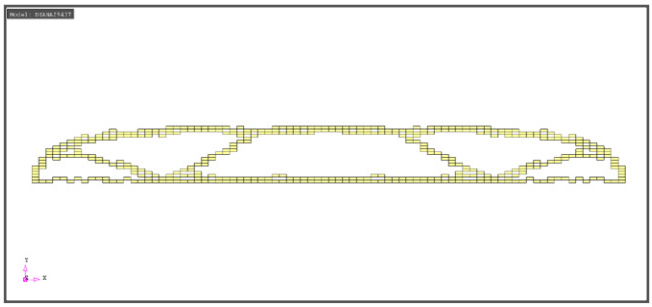
Mesh.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

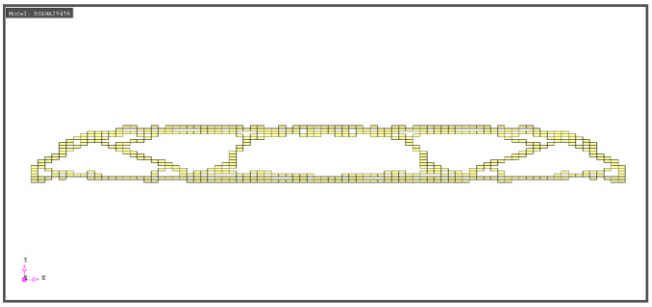
Performance function.



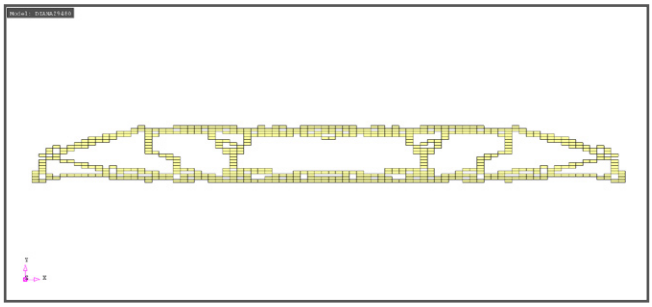
Best solution found in run 1.



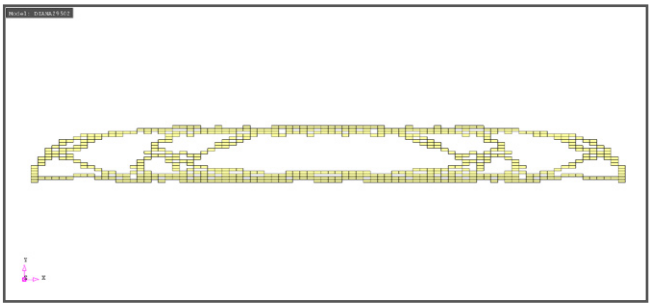
Best solution found in run 2.



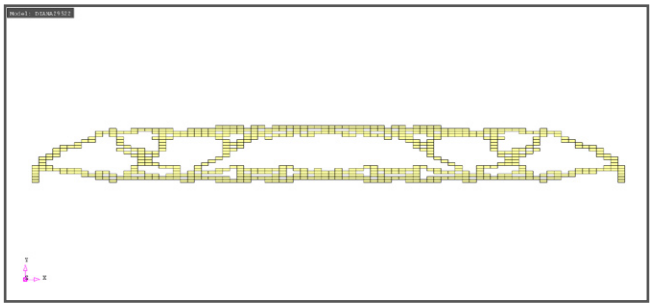
Best solution found in run 3.



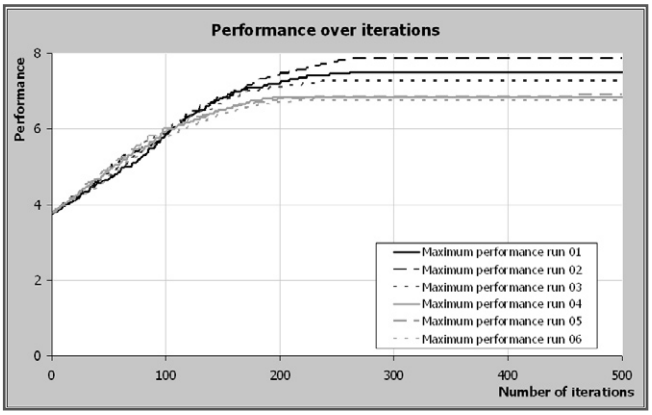
Best solution found in run 4.



Best solution found in run 5.



Best solution found in run 6.

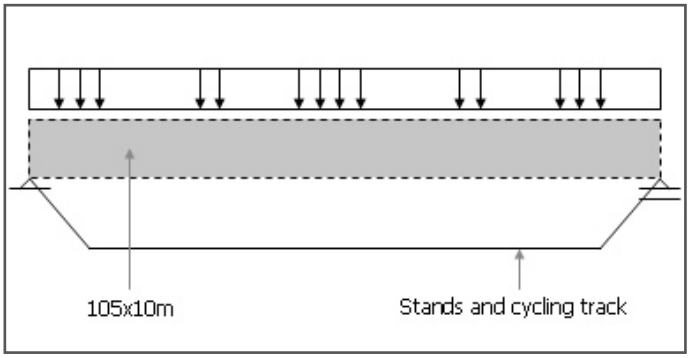


Performance chart of the runs.

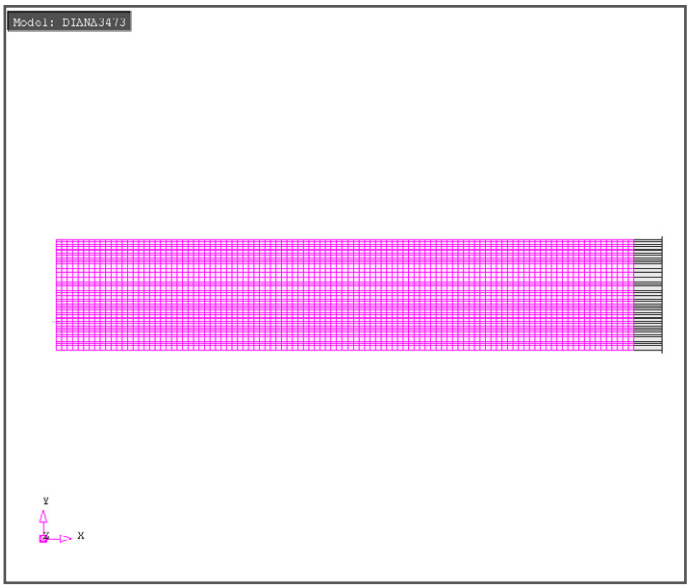
Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Dimensions	2	
Stress components	3	
Nodes	5406	
Elements	5250	
Integration points	4	
Iterations	500	
Ants	200	
Elitist ants	8	
Ant runs	100000	
RV initial	0.8	
RV maximum	0.999	
RV minimum	0.001	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m²
Load case ULS2	0.20	kN/m²
Load case SLS1	2.26	kN/m²
Load case SLS2	0.85	kN/m²
Preferred nr. of holes	2	
Hole width	1	m
Hole height	2	m
Run time (average)	135:39:43	hh:mm:ss
Run time (average)	488383	s
Average time per ant run	4.884	s

Job information.



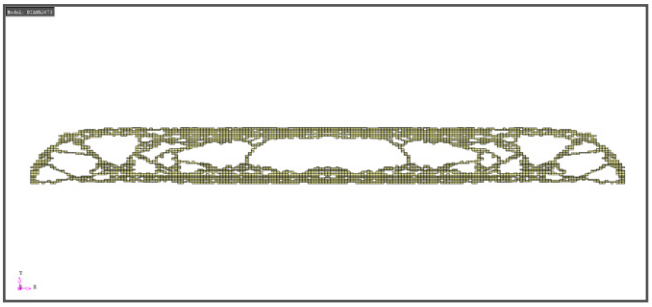
Design space and boundary conditions.



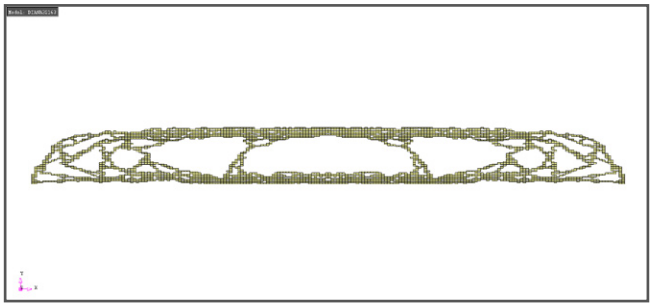
Mesh.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

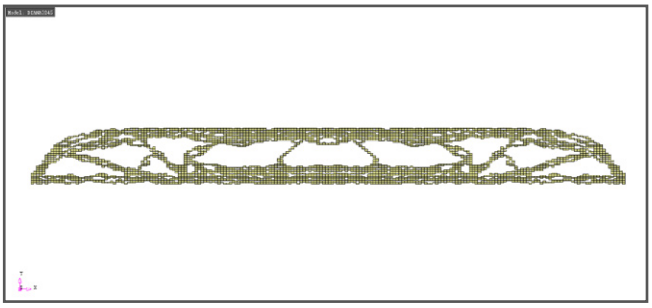
Performance function.



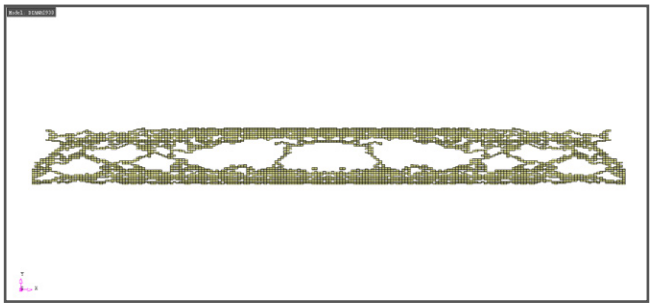
Best solution found in run 1.



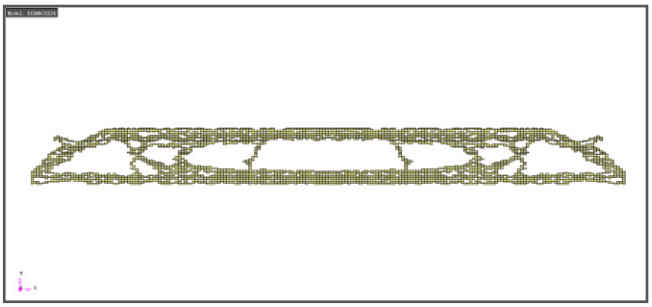
Best solution found in run 2.



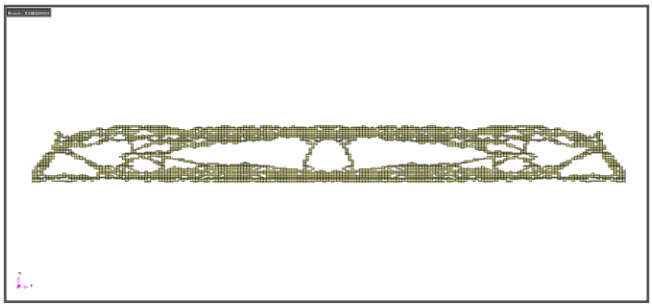
Best solution found in run 3.



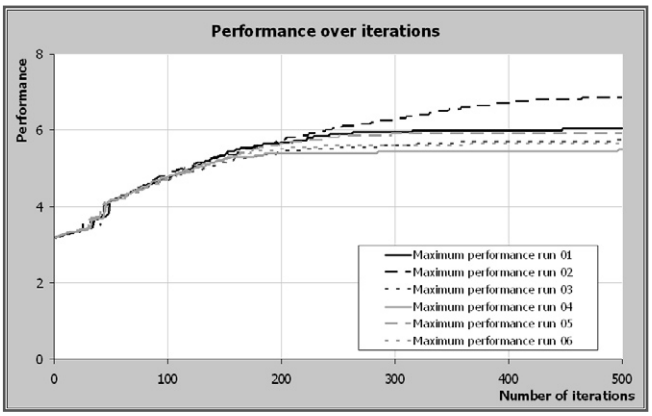
Best solution found in run 4.



Best solution found in run 5.



Best solution found in run 6.



Performance chart of the runs.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

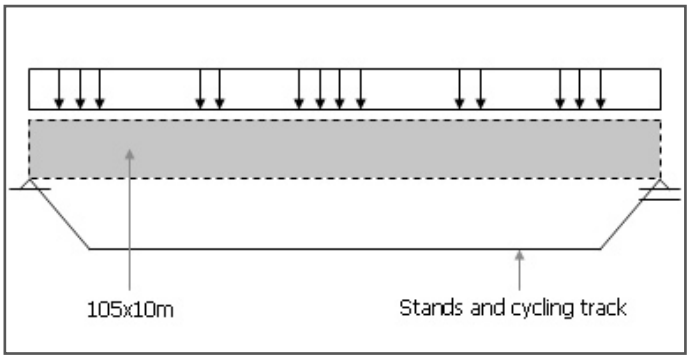
D.4.3 Complete model

Dimensions	2	
Stress components	3	
Nodes	1785	
Elements	1680	
Integration points	4	
Iterations	500	
Ants	200	
Elitist ants	8	
Ant runs	100000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m²
Load case ULS2	0.20	kN/m²
Load case SLS1	2.26	kN/m²
Load case SLS2	0.85	kN/m²
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	17:27:58	hh:mm:ss
Run time (average)	62878	s
Average time per ant run	0.629	s

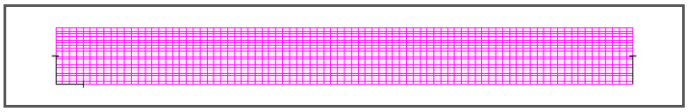
Job information.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

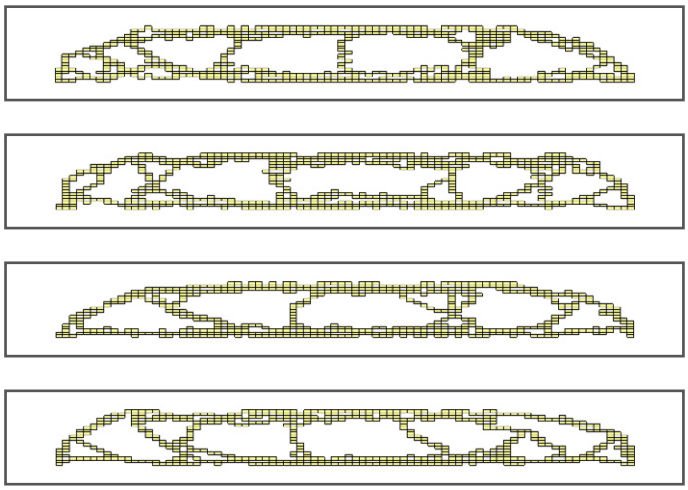
Performance function.



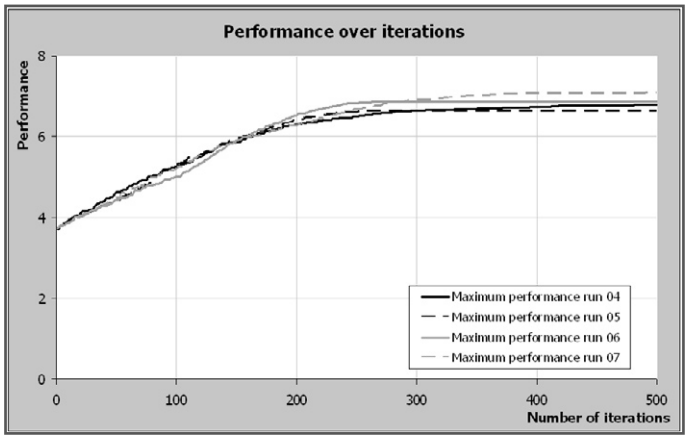
Design space and boundary conditions.



Mesh.



Best solutions found in runs 4, 5, 6 and 7.



Performance chart of the runs.

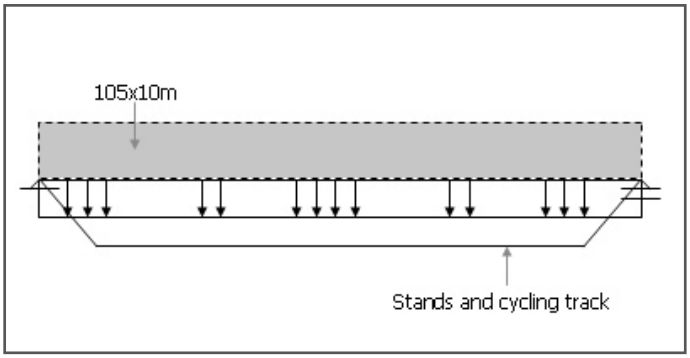
D.4.4 Load on bottom

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	500	
Ants	100	
Elitist ants	8	
Ant runs	50000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m^2
Load case ULS2	0.20	kN/m^2
Load case SLS1	2.26	kN/m^2
Load case SLS2	0.85	kN/m^2
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	3:33:54	hh:mm:ss
Run time (average)	12834	s
Average time per ant run	0.257	s

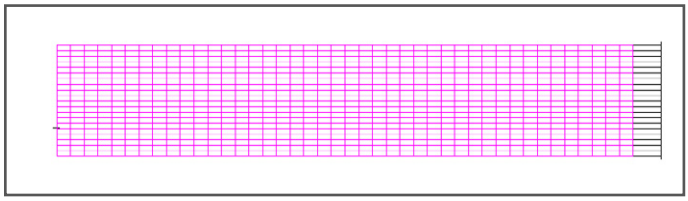
Job information.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

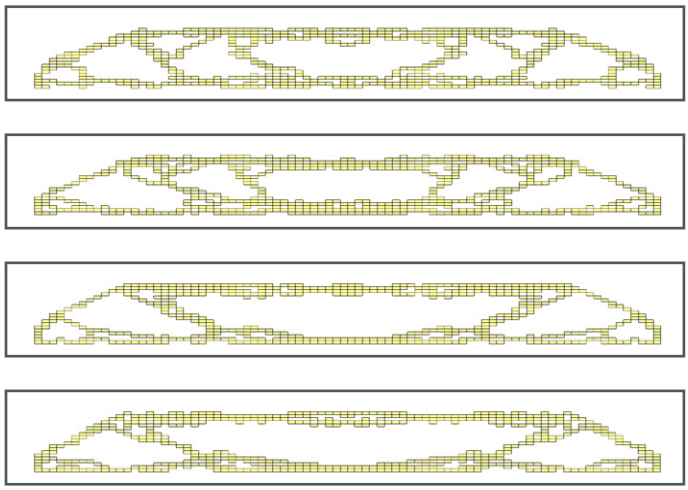
Performance function.



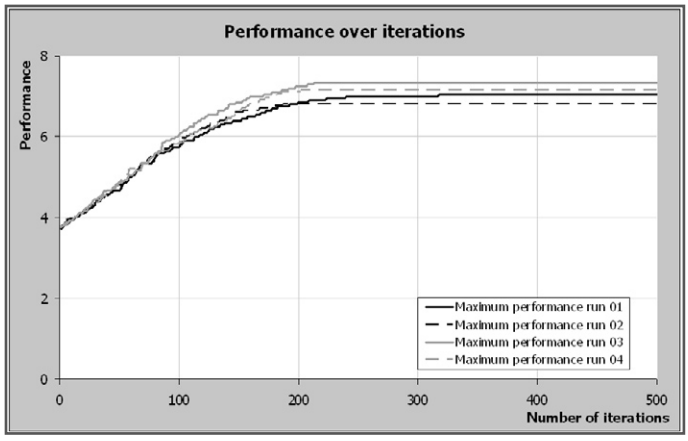
Design space and boundary conditions.



Mesh.



Best solutions found in runs 1, 2, 3 and 4.



Performance chart of the runs.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

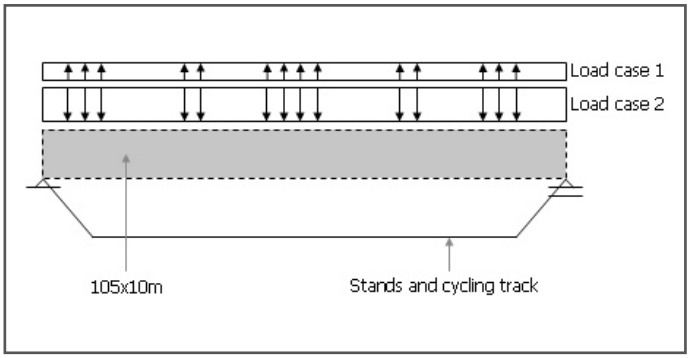
D.4.5 Up- and downward load

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	500	
Ants	100	
Elitist ants	8	
Ant runs	50000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.999	
Stress treshold 2	0.05	
Performance treshold 2	0.999	
Load case ULS1	3.00	kN/m²
Load case ULS2	-0.26	kN/m²
Load case SLS1	2.26	kN/m²
Load case SLS2	0.35	kN/m²
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	3:38:05	hh:mm:ss
Run time (average)	13085	s
Average time per ant run	0.262	s

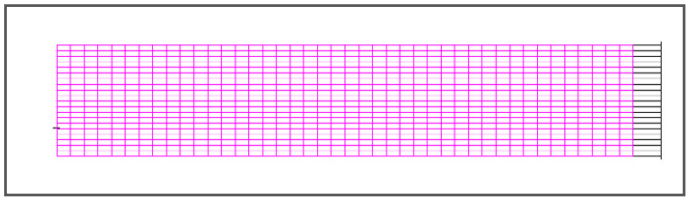
Job information.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

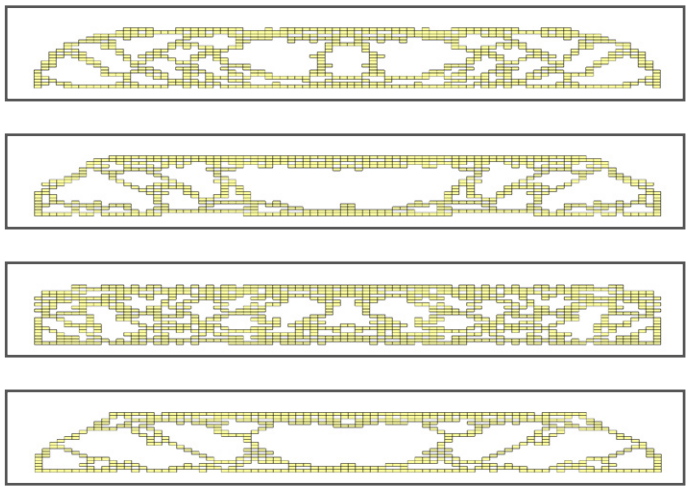
Performance function.



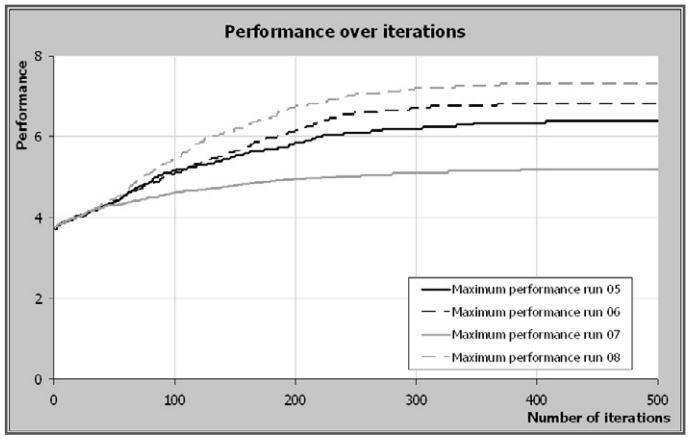
Design space and boundary conditions.



Mesh.



Best solutions found in runs 5, 6, 7 and 8.



Performance chart of the runs.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

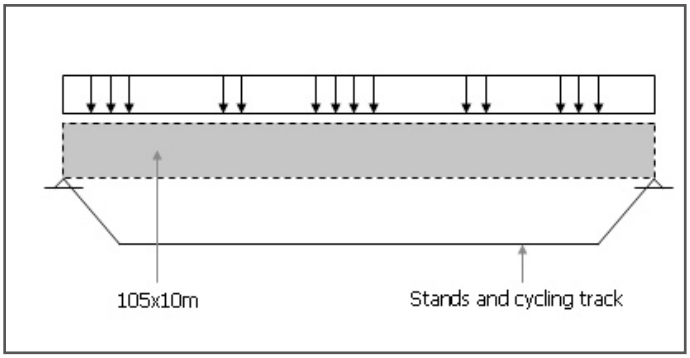
D.4.6 Pinned model

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	500	
Ants	100	
Elitist ants	8	
Ant runs	50000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m^2
Load case ULS2	0.20	kN/m^2
Load case SLS1	2.26	kN/m^2
Load case SLS2	0.85	kN/m^2
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	3:50:33	hh:mm:ss
Run time (average)	13833	s
Average time per ant run	0.277	s

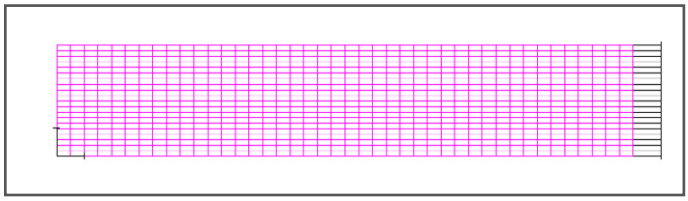
Job information.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

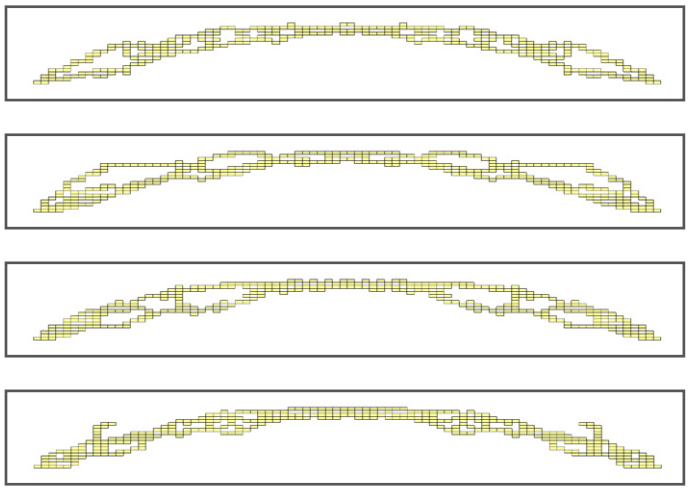
Performance function.



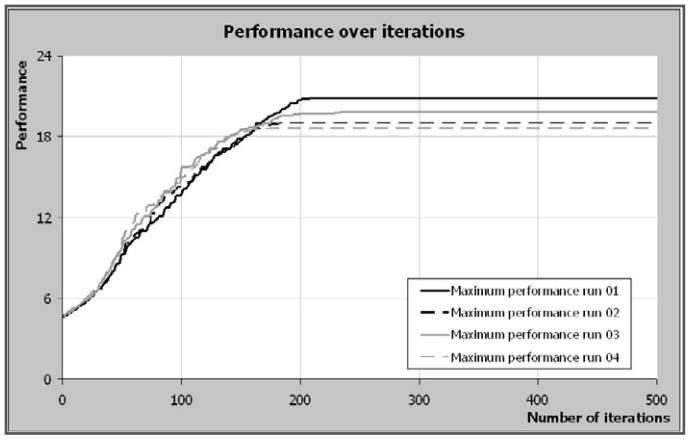
Design space and boundary conditions.



Mesh.



Best solutions found in runs 1, 2, 3 and 4.



Performance chart of the runs.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

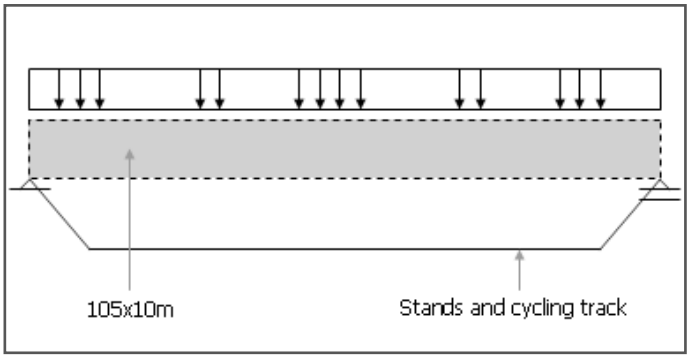
D.4.7 Ordinary concrete

Dimensions	2	
Stress components	3	
Nodes	903	
Elements	840	
Integration points	4	
Iterations	500	
Ants	100	
Elitist ants	8	
Ant runs	50000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	2.50E+07	Pa
Fy;min	-2.93E+07	Pa
Ft	2.15E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	16.5	kN/m²
Load case ULS2	14.3	kN/m²
Load case SLS1	13.6	kN/m²
Load case SLS2	12.1	kN/m²
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	3:51:15	hh:mm:ss
Run time (average)	13875	s
Average time per ant run	0.278	s

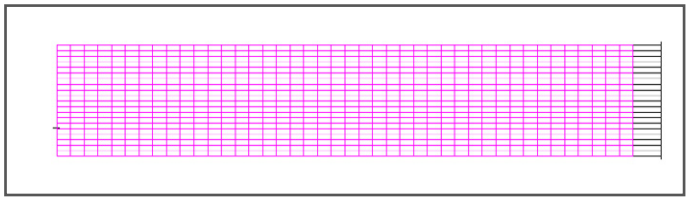
Job information.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

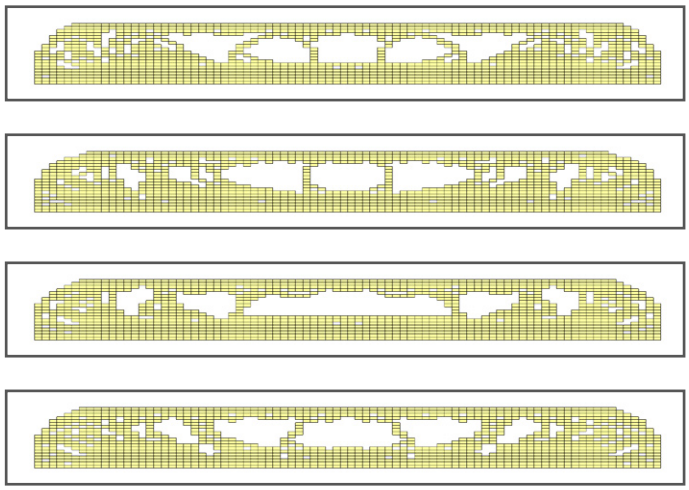
Performance function.



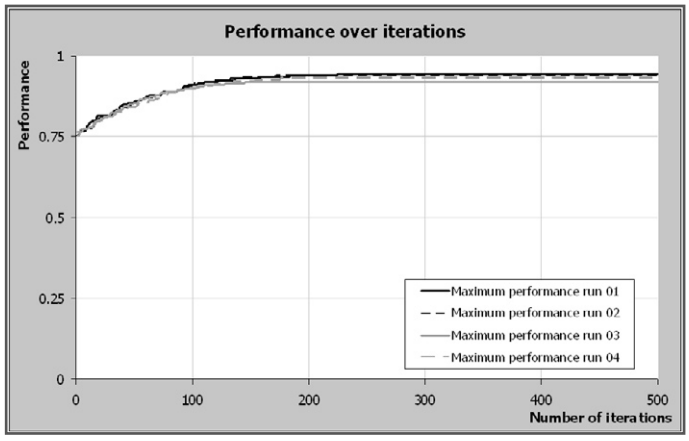
Design space and boundary conditions.



Mesh.



Best solutions found in runs 1, 2, 3 and 4.



Performance chart of the runs.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

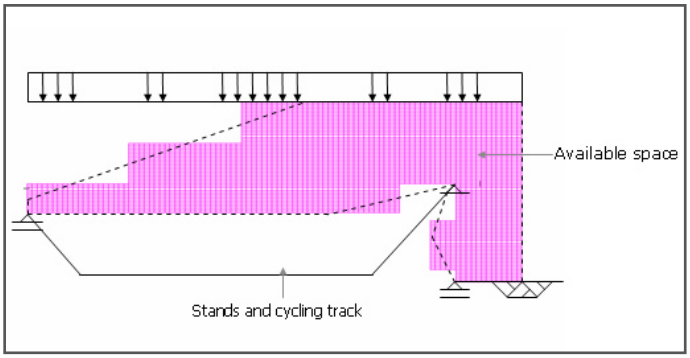
D.4.8 Enlarged search space

Dimensions	2	
Stress components	3	
Nodes	4611	
Elements	4406	
Integration points	4	
Iterations	500	
Ants	200	
Elitist ants	8	
Ant runs	100000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.5	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m^2
Load case ULS2	0.20	kN/m^2
Load case SLS1	2.26	kN/m^2
Load case SLS2	0.85	kN/m^2
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	76:13:20	hh:mm:ss
Run time (average)	274400	s
Average time per ant run	2.744	s

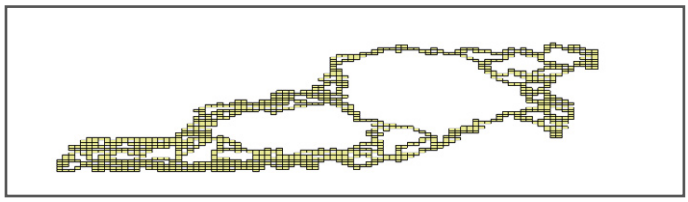
Job information.

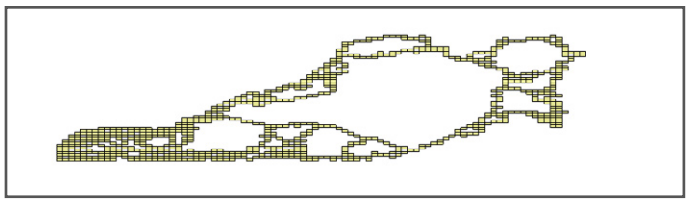
$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

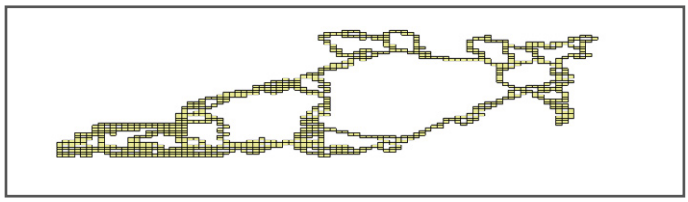
Performance function.

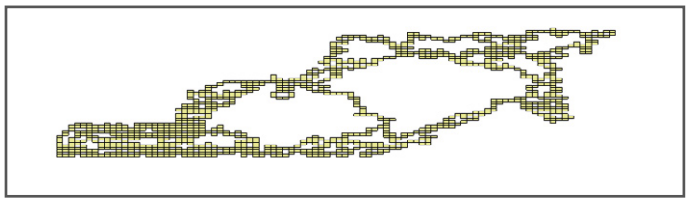


Design space, boundary conditions and mesh.

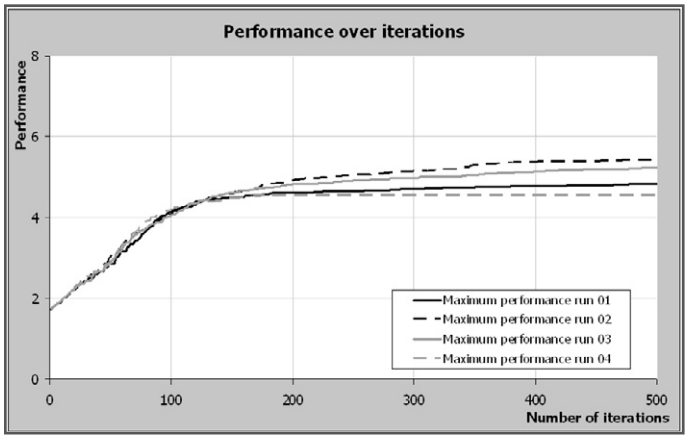








Best solutions found in runs 1, 2, 3 and 4.



Performance chart of the runs.

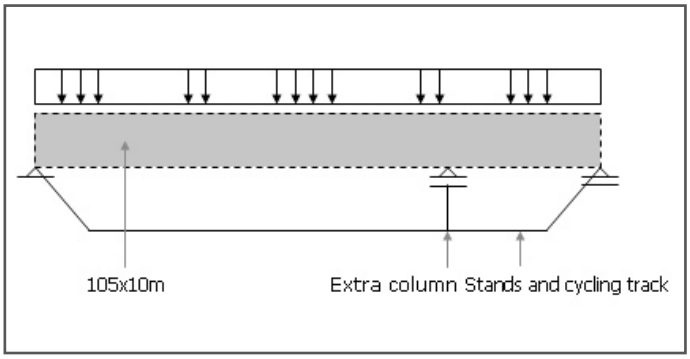
D.4.9 Extra support

Dimensions	2	
Stress components	3	
Nodes	1785	
Elements	1680	
Integration points	4	
Iterations	500	
Ants	200	
Elitist ants	8	
Ant runs	100000	
RV initial	0.8	
RV maximum	1	
RV minimum	0	
Rho	0.2	
Fy;max	8.16E+07	Pa
Fy;min	-7.65E+07	Pa
Ft	4.60E+06	Pa
Allowed deflection	0.35	m
Stress treshold 1	0.01	
Performance treshold 1	0.9	
Stress treshold 2	0.05	
Performance treshold 2	0.9	
Load case ULS1	3.00	kN/m²
Load case ULS2	0.20	kN/m²
Load case SLS1	2.26	kN/m²
Load case SLS2	0.85	kN/m²
Preferred nr. of holes	2	
Hole width	1.25	m
Hole height	2	m
Run time (average)	7:43:32	hh:mm:ss
Run time (average)	60565	s
Average time per ant run	0.606	s

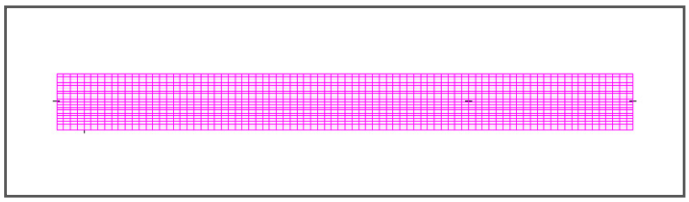
Job information.

$$perf_j = \begin{cases} \sim 0 & if\ u_{max} > u_{allowable} \\ \sim 0 & if\ \sigma_{max} > \sigma_{allowable;max} \\ \sim 0 & if\ \sigma_{min} < \sigma_{allowable;min} \\ \sim 0 & if\ \left(\sigma_{i;max} - \sigma_{i;min}\right) > \left(\sigma_{allowable;max} - f_t\right) \\ \dfrac{1000}{2V + 0.05S + 10PV_p + 20H} & else \end{cases}$$

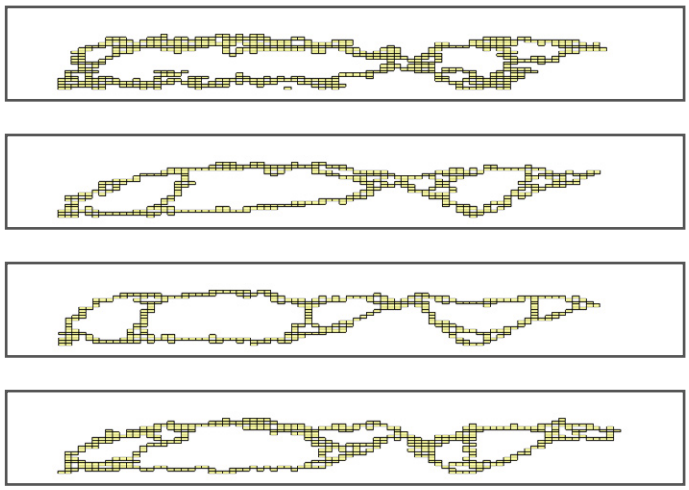
Performance function.



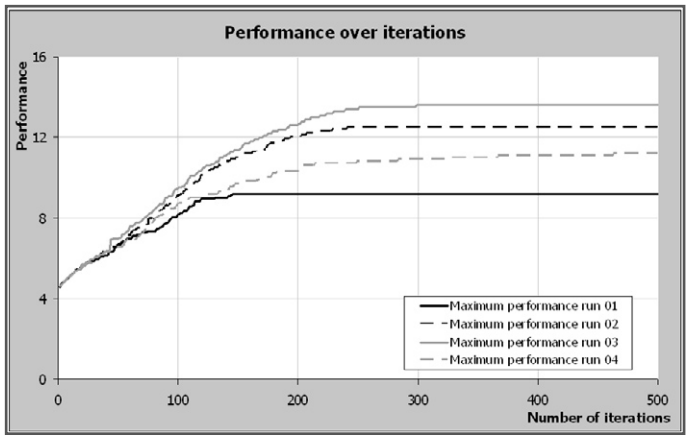
Design space and boundary conditions.



Mesh.



Best solutions found in runs 1, 2, 3 and 4.

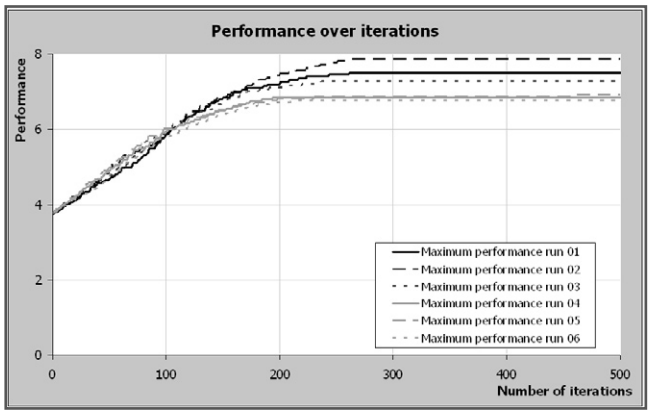


Performance chart of the runs.

D.5 Omnisport roof design runs

D.5.1 Coarse mesh

Information is additional to the information in Appendix D.4.1, where basic run data, parameters, mesh, boundary conditions, the used performance function et cetera can be found.

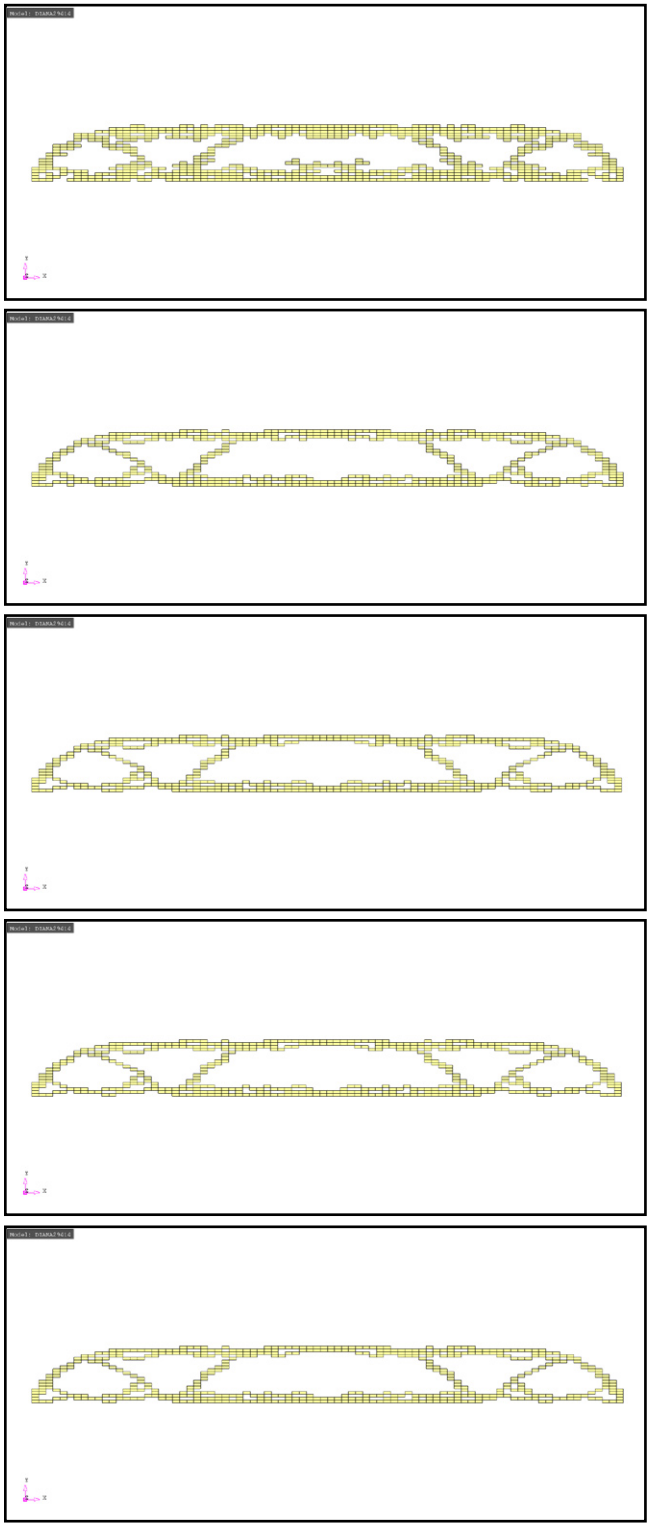


Performance chart of the runs.

Run 1

Run time	4:41:32	hh:mm:ss
Run time	16892	s
Time per ant run	0.338	s

Run times.



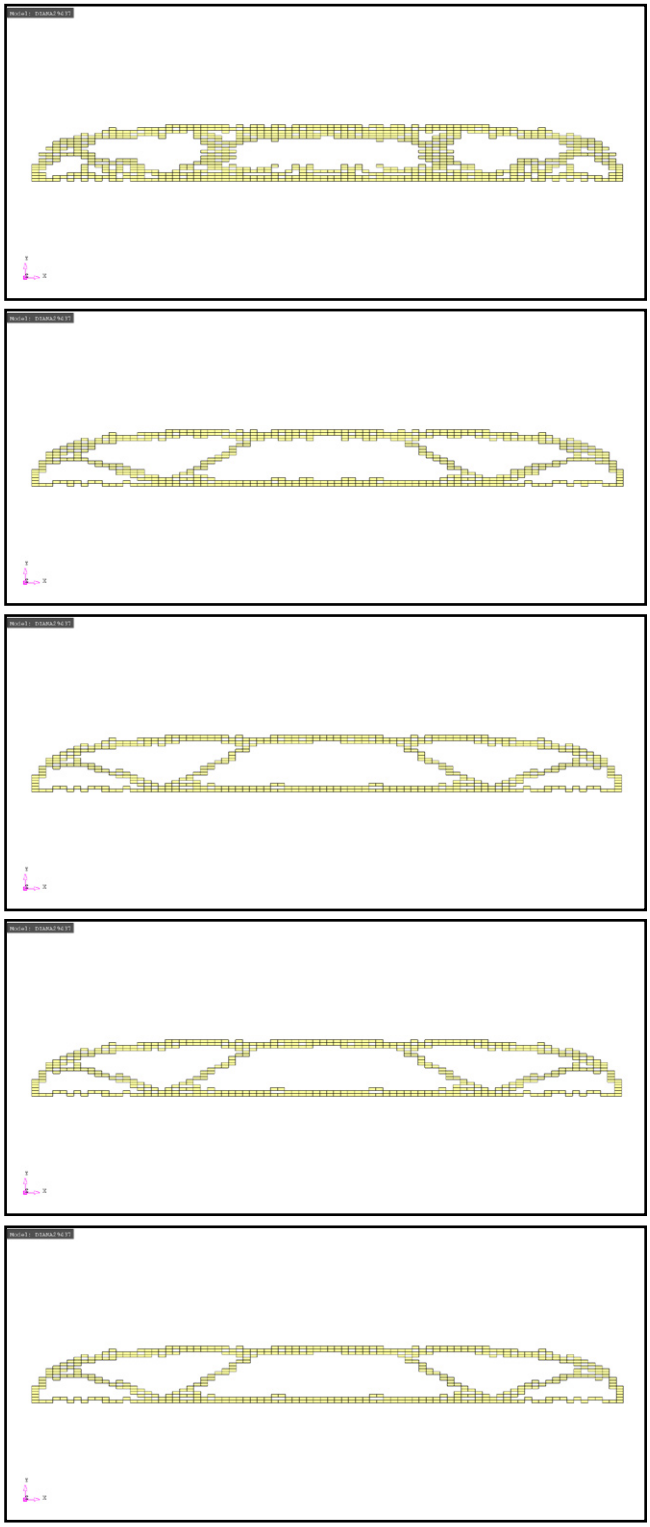
Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 2

Run time	4:41:21	hh:mm:ss
Run time	16881	s
Time per ant run	0.338	s

Run times.

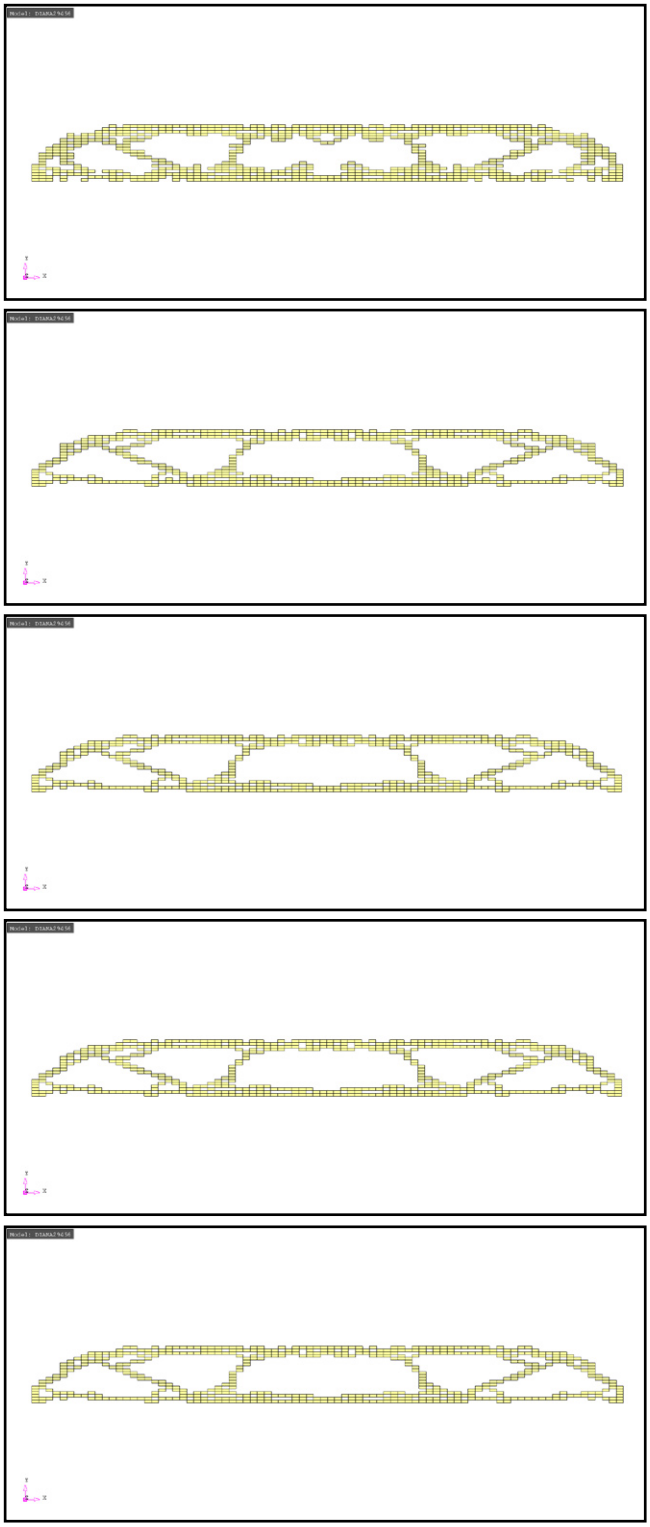


Best solutions found after 100, 200, 300, 400 and 500 iterations.

Run 3

Run time	4:40:33	hh:mm:ss
Run time	16833	s
Time per ant run	0.337	s

Run times.



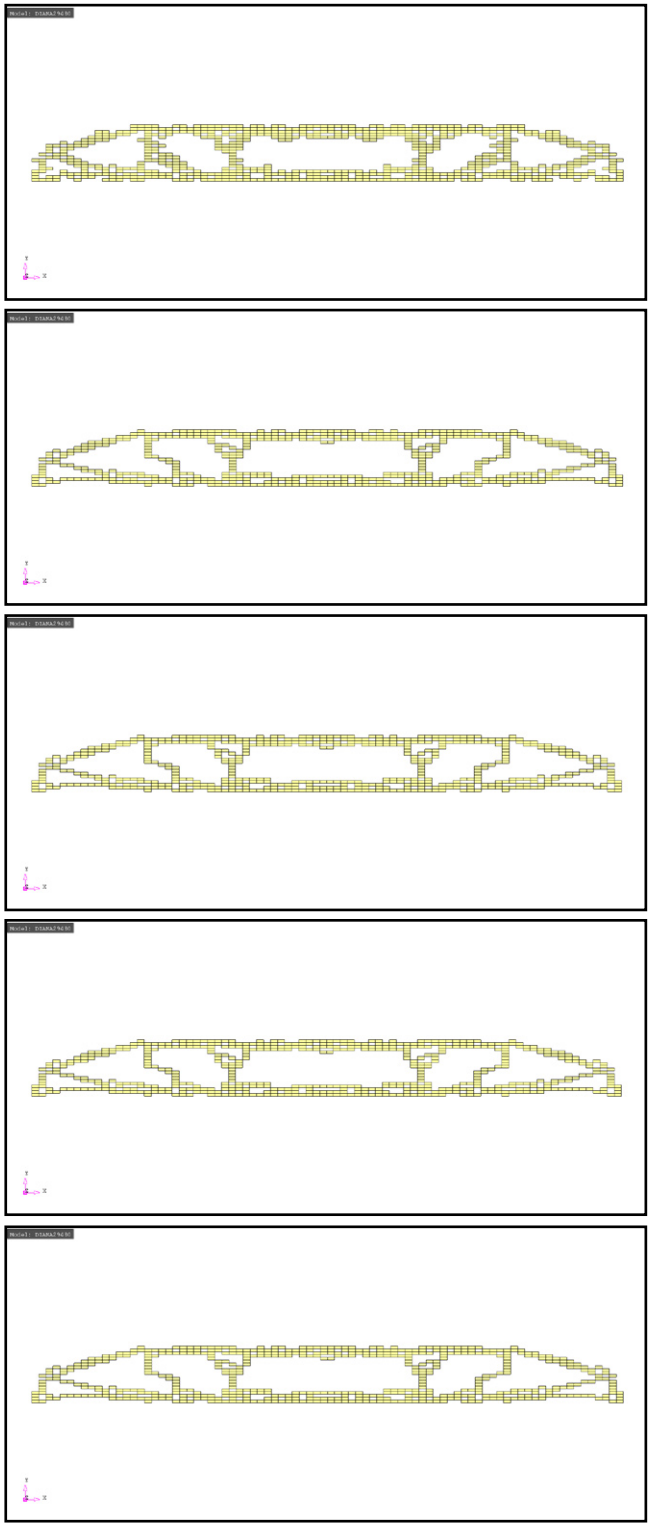
Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 4

Run time	4:41:52	hh:mm:ss
Run time	16912	s
Time per ant run	0.338	s

Run times.

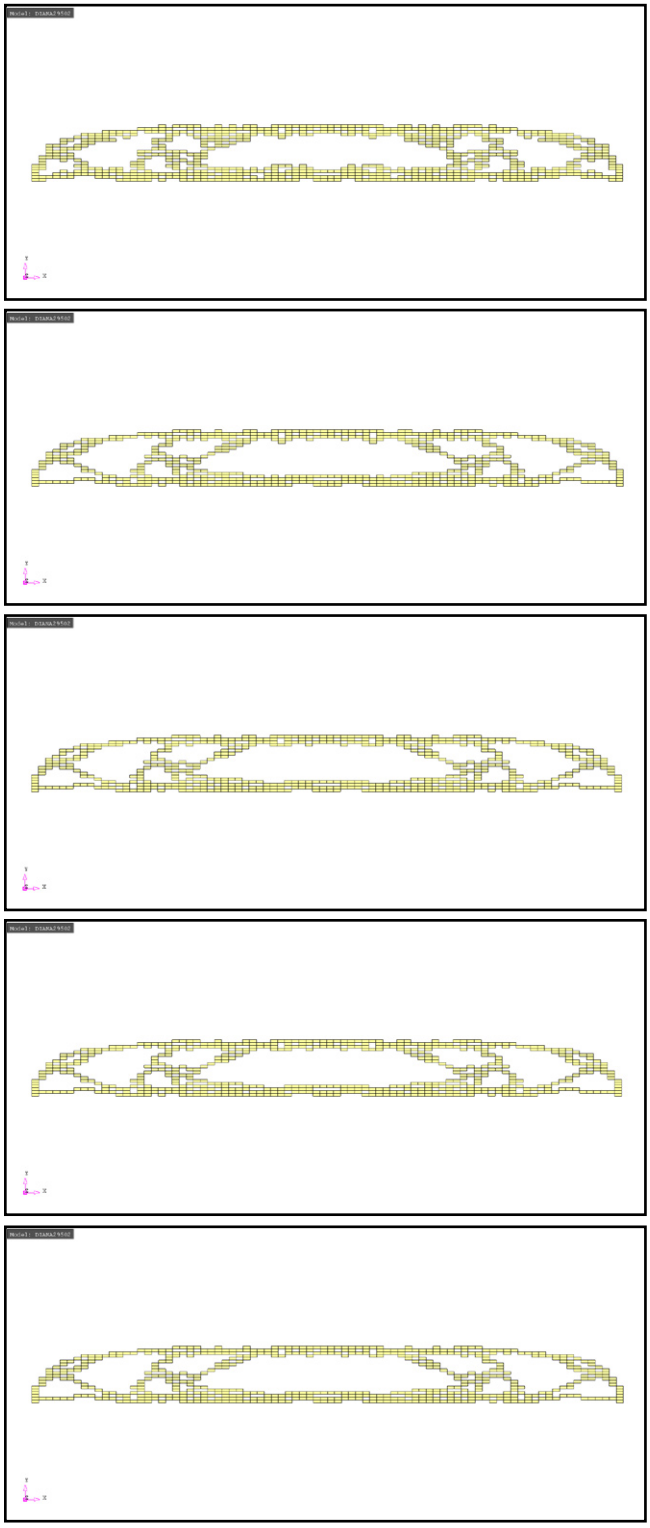


Best solutions found after 100, 200, 300, 400 and 500 iterations.

Run 5

Run time	4:42:25	hh:mm:ss
Run time	16945	s
Time per ant run	0.339	s

Run times.



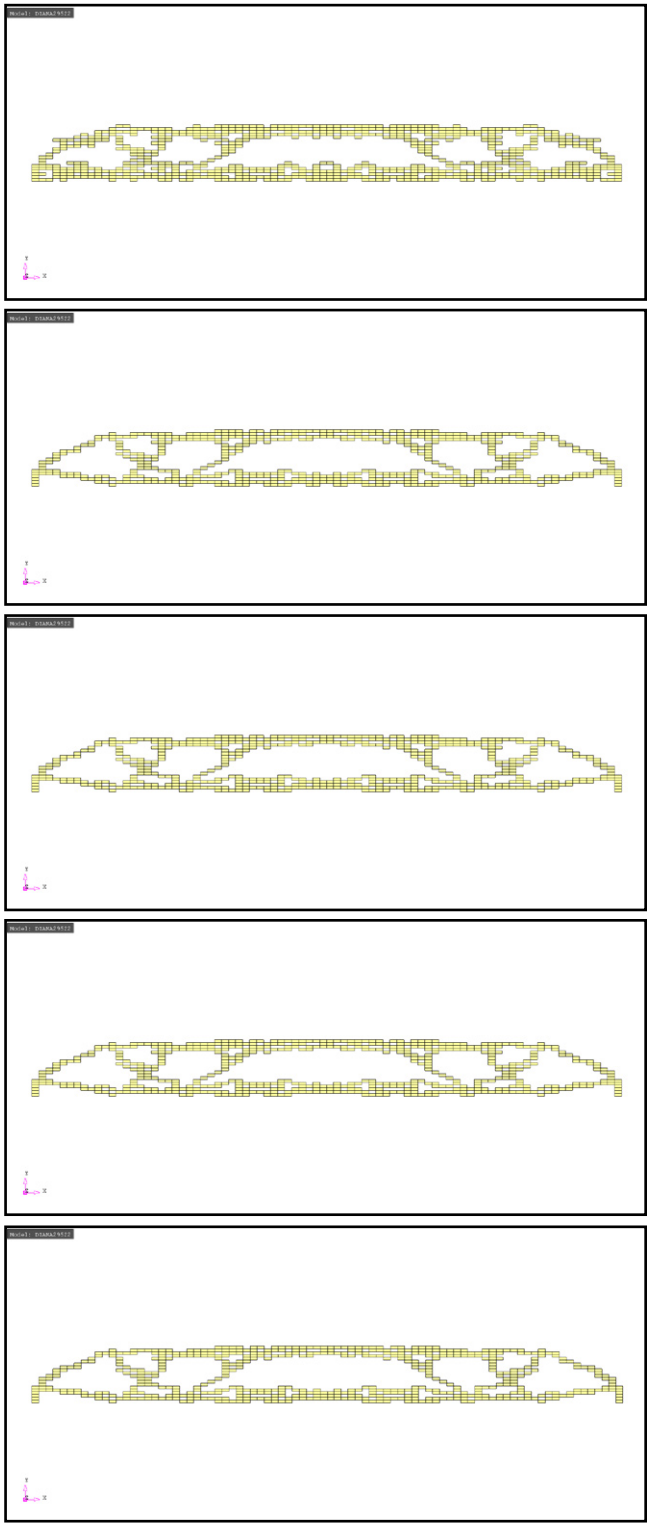
Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 6

Run time	4:41:07	hh:mm:ss
Run time	16867	s
Time per ant run	0.339	s

Run times.

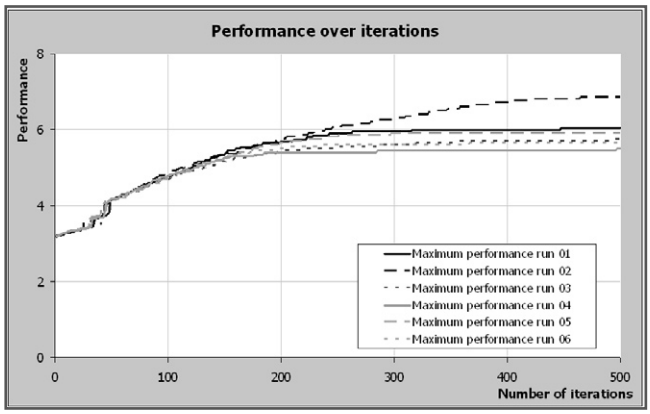


Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

D.5.2 Fine mesh

Information is additional to the information in Appendix D.4.2, where basic run data, parameters, mesh, boundary conditions, the used performance function et cetera can be found.

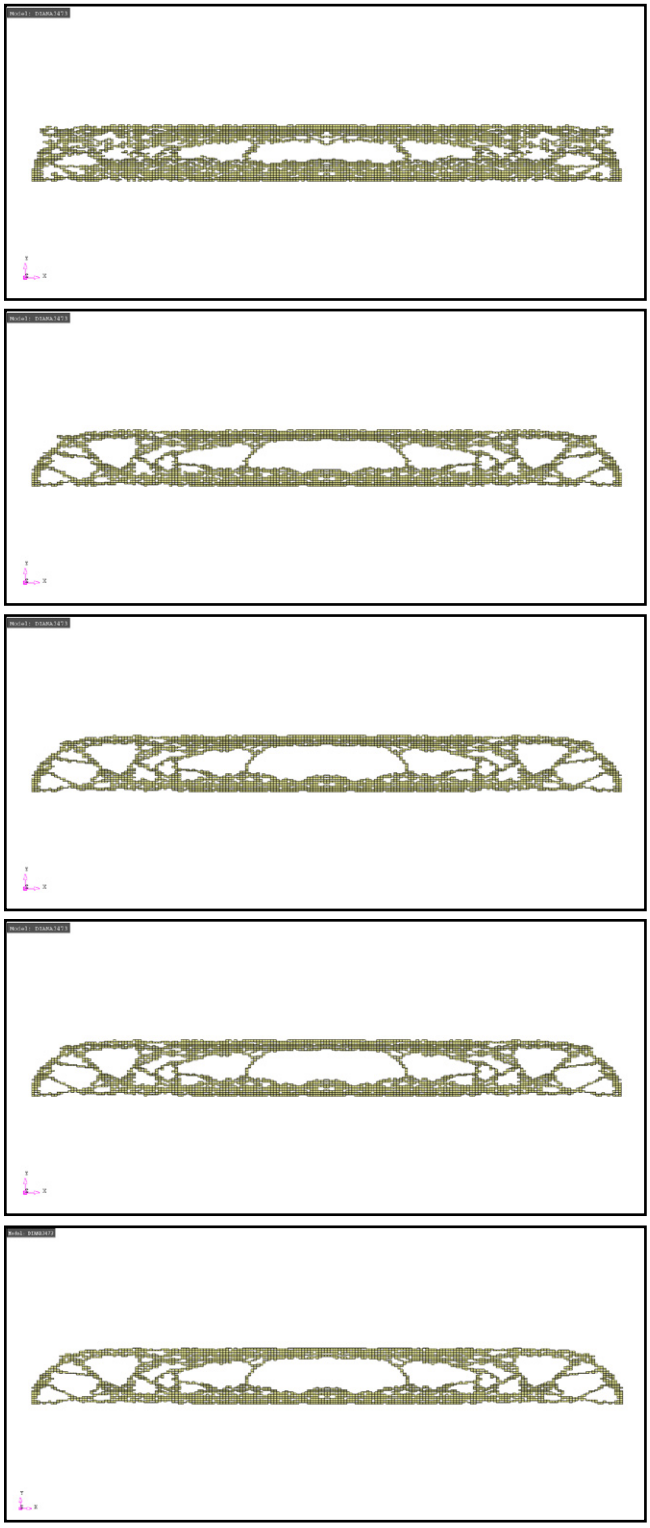


Performance chart of the runs.

Run 1

Run time	142:57:00	hh:mm:ss
Run time	514620	s
Time per ant run	5.146	s

Run times.



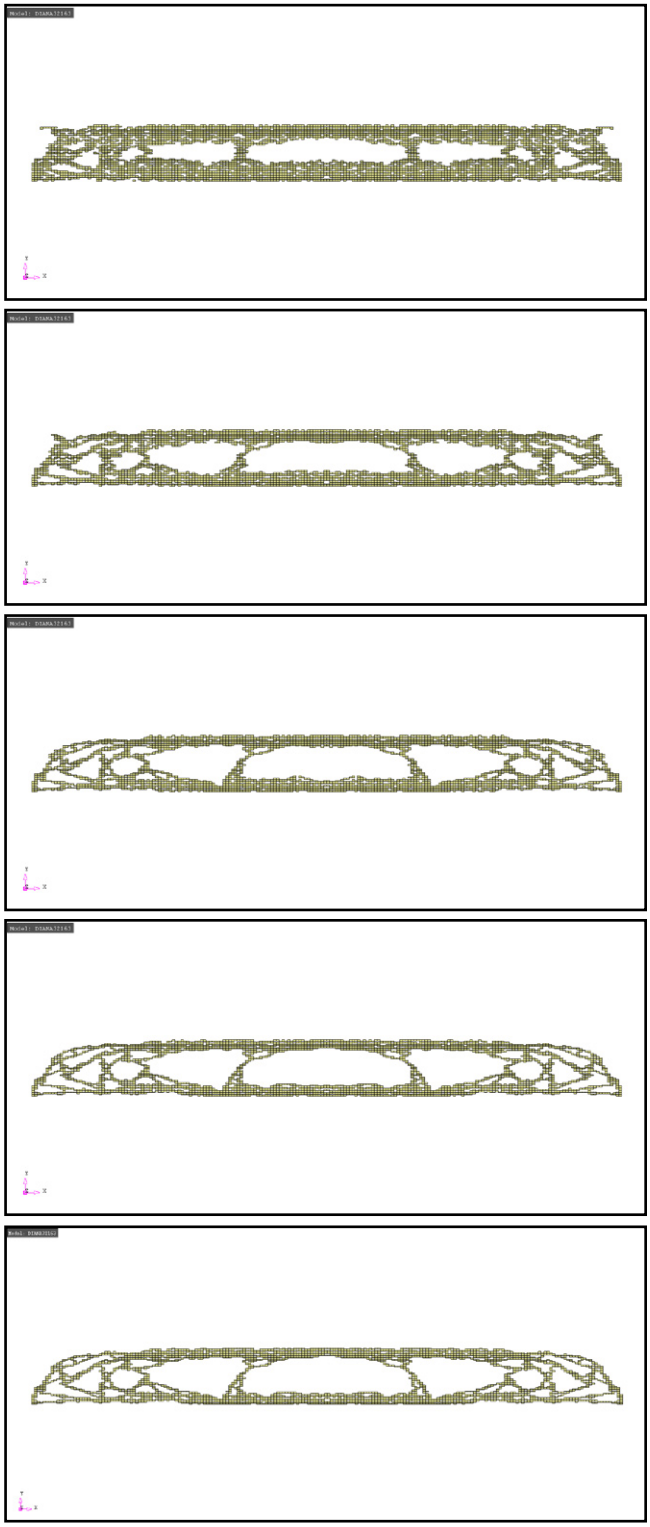
Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 2

Run time	137:29:12	hh:mm:ss
Run time	494952	s
Time per ant run	4.950	s

Run times.

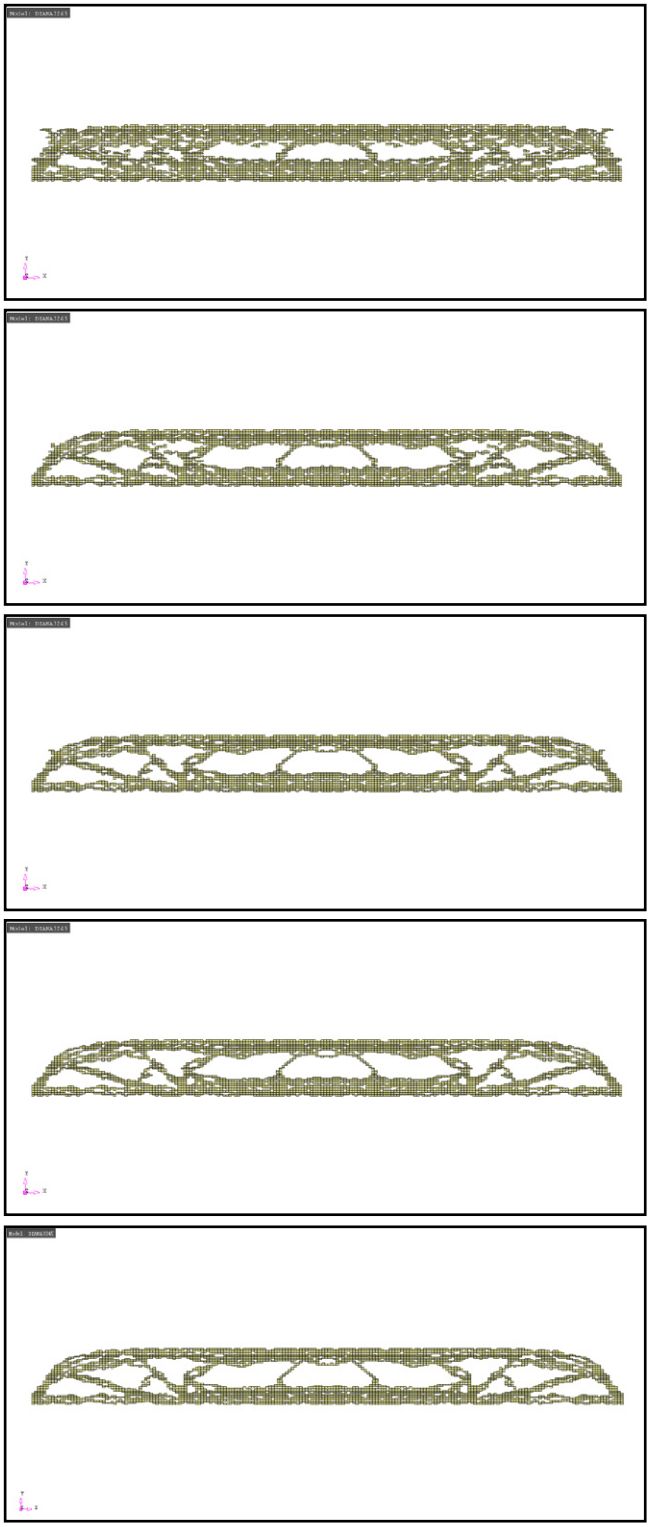


Best solutions found after 100, 200, 300, 400 and 500 iterations.

Run 3

Run time	142:42:41	hh:mm:ss
Run time	513761	s
Time per ant run	5.138	s

Run times.



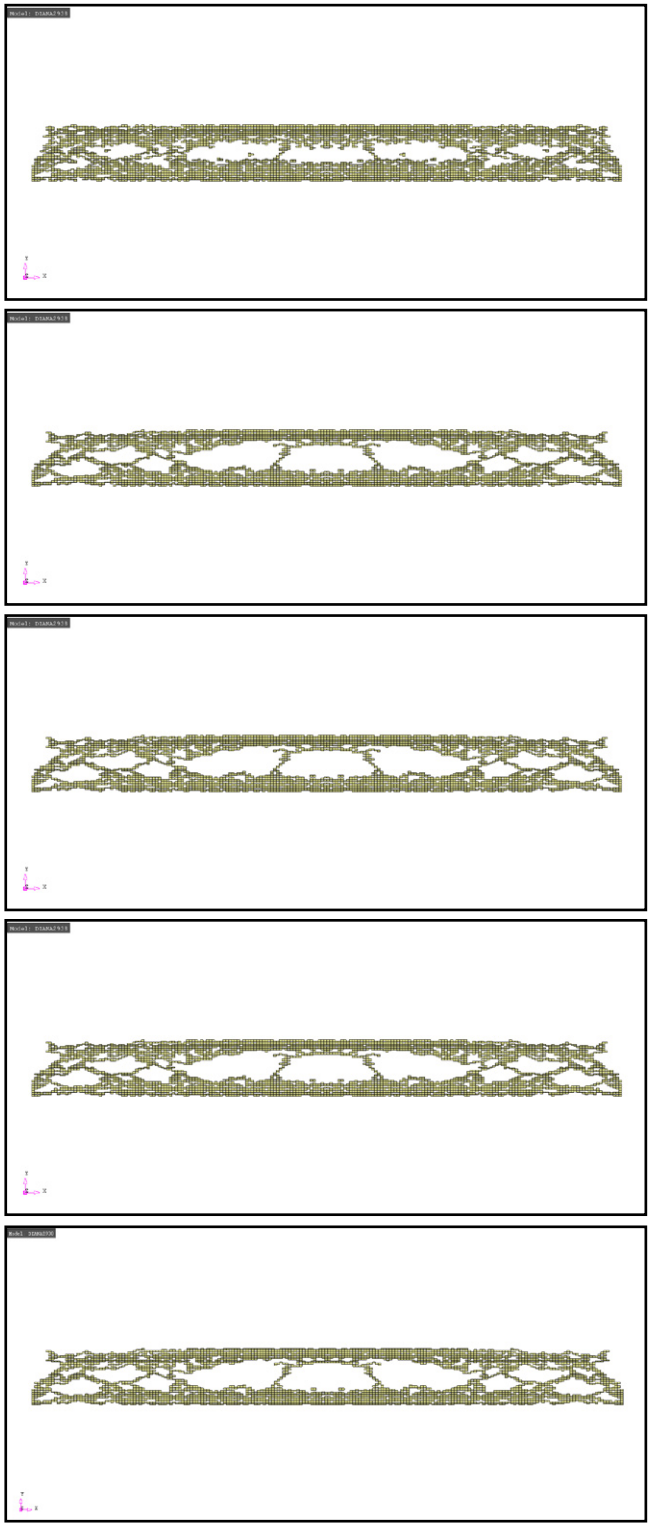
Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 4

Run time	141:50:21	hh:mm:ss
Run time	510621	s
Time per ant run	5.106	s

Run times.

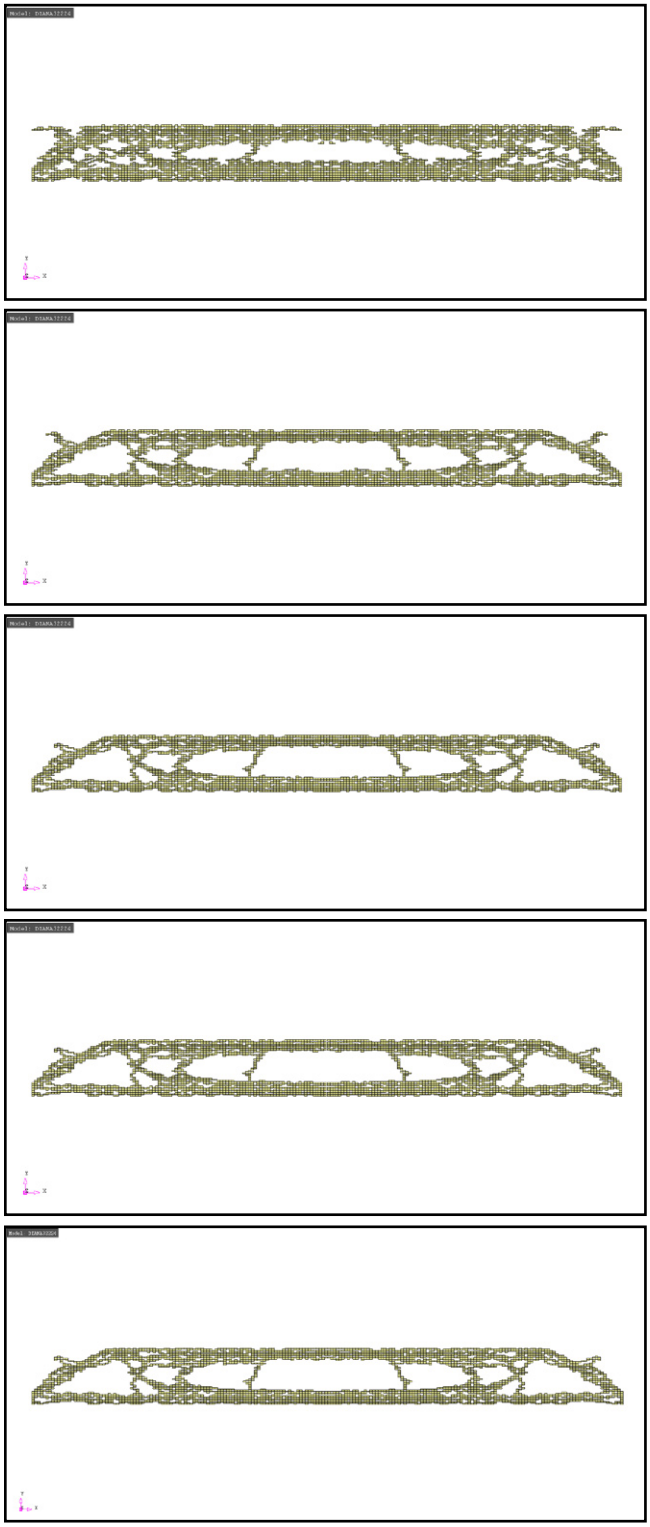


Best solutions found after 100, 200, 300, 400 and 500 iterations.

Run 5

Run time	131:00:45	hh:mm:ss
Run time	471645	s
Time per ant run	4.716	s

Run times.



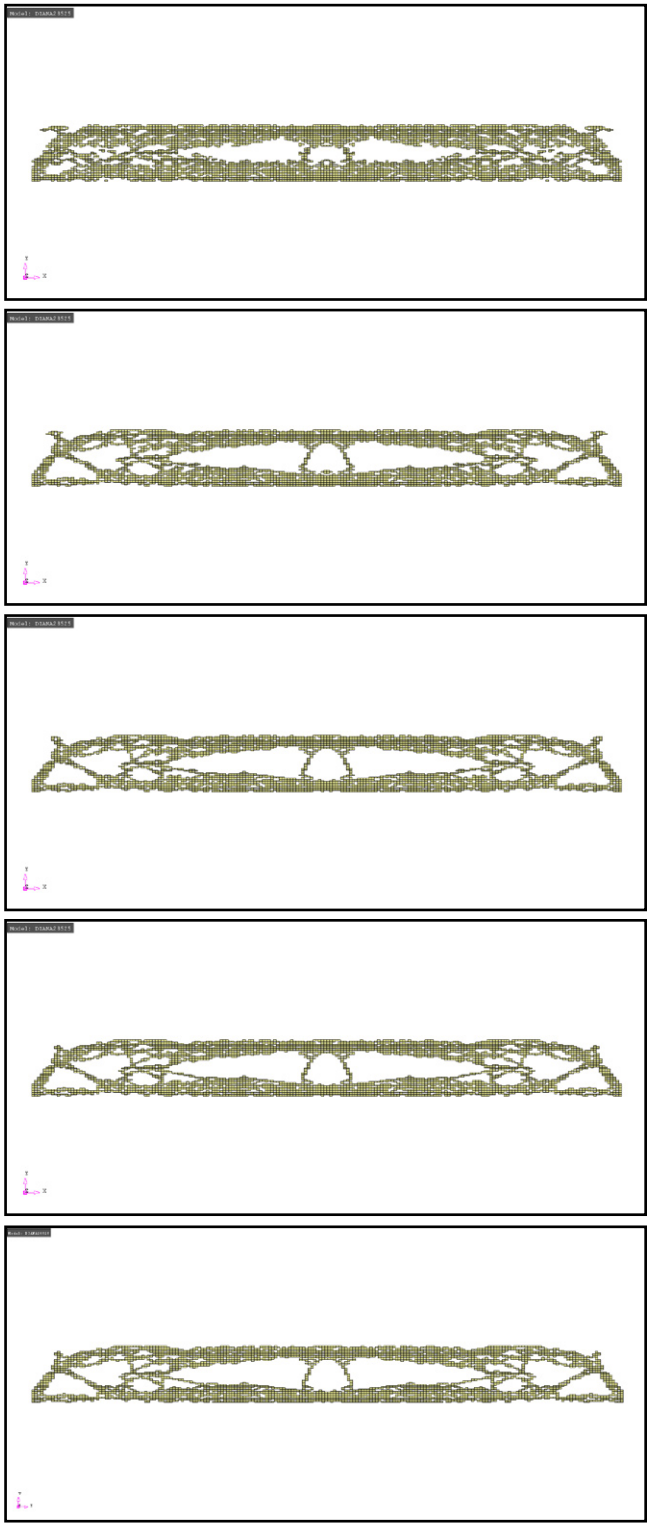
Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

Run 6

Run time	117:58:41	hh:mm:ss
Run time	424721	s
Time per ant run	4.247	s

Run times.



Best solutions found after 100, 200, 300, 400 and 500 iterations.

Visual output is mirrored along the line of symmetry; only half the problem is analysed.

APPENDIX E DESIGN

E.1 Self weight estimate

CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. MF Afstuderen	blz. no.
		berek. no. Self weight estimation	

Iterative calculation to determine the self weight of a structural element spanning the biggest span.

Data:

Stress in concrete	=	50	N/mm² (conservative assumpti
Structural depth:	=	9	m (assumption)
Span	=	105	m
Centre to centre distance	=	12	m
Volumic mass	=	28	kN/m³
Snow load	=	0.56	kN/m²
Dead load	=	0.80	kN/m²

Calculations below are the result of an iterative process, assuming the element will be a truss.

Surface on top and bottom : $0{,}103\ m^2$

Extra surface for diagonals and as a safety : $1{,}5 \cdot 0{,}103 = 0{,}155\ m^2$

Total cross-sectional surface : $2 \cdot 0{,}103 + 0.155 = 0{,}362\ m^2$

Self weight per metre span : $28 \cdot 0{,}362 = 10{,}1\ kN/m'$

Self weight per sq. m. roof : $10{,}1 / 12 = 0{,}84\ kN/m^2$

Factorised downward load, ULS: $1{,}2(0{,}84+0{,}80) + 1{,}5 \cdot 0{,}56 = 2{,}81\ kN/m^2$

Maximum bending moment $\frac{1}{8} \cdot (12 \cdot 2{,}81) \cdot 105^2 = 46.530\ kNm$

Force in upper and lower chord: $46530 / 9 = 5170\ kN$

Area needed in top and bottom: $5170000 / 50 = 103400\ mm^2 = 0.103\ m^2$

According to this calculation, the self weight of the structural element will be around $0{,}84\ kN/m^2$

E.2 Member dimensions calculations

CORSMIT RAADGEVEND INGENIEURSBUREAU BV	Postbus 208 2280 AE Rijswijk (ZH) tel. 070-394 93 05 fax 070-394 13 96	werk no. MF Afstuderen berek. no. dimensioneren staven	blz. no. 1/4

Member	Force	Buckling length
A	-2800 kN	10.4 m
B	-4000 kN	29.2 m
C	-4800 kN	30.0 m
D	+1600 kN	-
E	-1100 kN	17.1 m
F	0 kN	-
G	+2400 kN	-
H	+4800 kN	-

Calculations for buckling are based on [Wagemans 2003].

$i_y = \sqrt{\frac{I_y}{A}}$

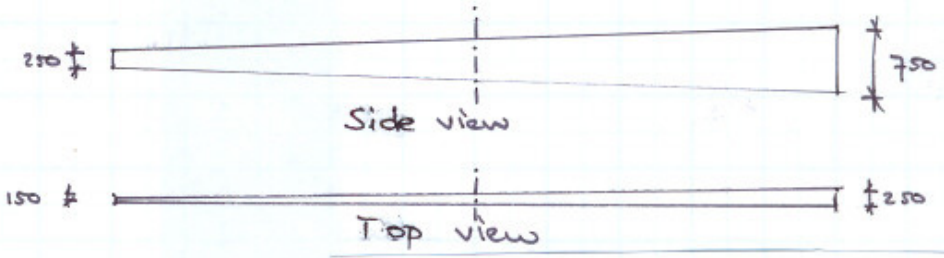
$i_z = \sqrt{\frac{I_z}{A}}$

$\sigma_{bcu} = 102$ MPa

$\sigma_{c;p} = 67$ MPa

Member A.

Tapered:



Cross-section at mid span:

$A = 50{,}000$ mm^2

$I_{yy} = 1.54 \cdot 10^9$ mm^4

$i_y = 176$ mm

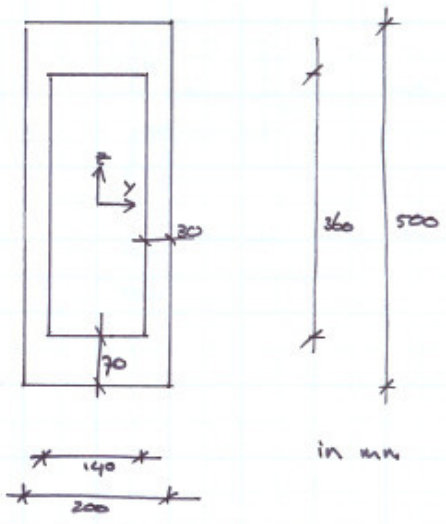
$\lambda = \frac{10400}{176} = 59$

$k = 0.75$

$\sigma = \frac{2080000}{50000} = 56$ MPa

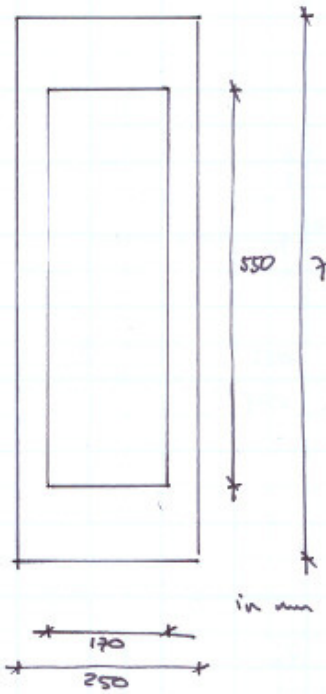
$= 0.55\ \sigma_{bcu}$

O.K.



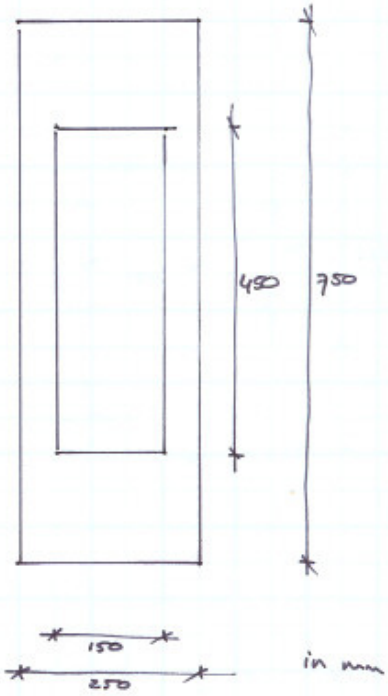
Member B

$A = 94.000 \text{ mm}^2$
$I_{yy} = 6.4 \cdot 10^9 \text{ mm}^4$
$i_y = 262 \text{ mm}$
$\lambda = \frac{29400}{262} = 112$
$k = 0.45$
$\sigma = \frac{4000}{94} = 43 \text{ MPa}$
$= 0{,}42\ \sigma_{bcu}$
O.K.



Member C

$A = 120.000 \text{ mm}^2$
$I_{yy} = 7.7 \cdot 10^9 \text{ mm}^4$
$i_y = 252 \text{ mm}$
$\lambda = \frac{30000}{252} = 119$
$k = 0.40$
$\sigma = \frac{4800}{120} = 40 \text{ MPa}$
$= 0.39\ \sigma_{bcu}$
O.K.

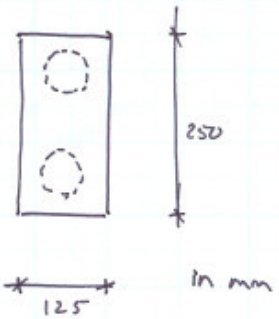


Member D

$F = 1600 \text{ kN}$
$\sigma_{c;p} = 67 \text{ MPa}$
$A_c = 24000 \text{ mm}^2$
$A_p = \frac{1600000}{1250\ 1120} = 1428 \text{ mm}^2 = 10 \times \phi\ 15.7$

2 x tendons with 5 strands
VSL-VMA type EC : ϕ67

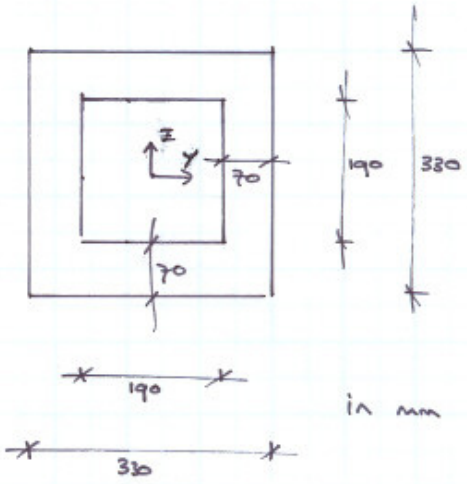
$A_{tendon} = 3530 \text{ mm}^2$
$A = 24000 + 2 \cdot 3530 = 31060 \text{ mm}^2$
$= 250 \times 125 \text{ mm}$



Member E

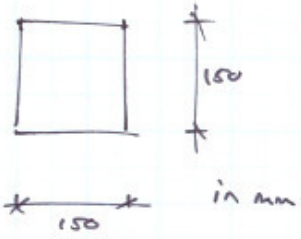
$A = 72800 \text{ mm}^2$
$I_{yy} = I_{zz} = 880 \cdot 10^6 \text{ mm}^4$
$i_y = i_z = 110 \text{ mm}$
$\lambda = \frac{13100}{110} = 156$
$k = 0{,}25$
$\sigma = \frac{1600}{72.8} = 22 \text{ MPa} = 0{,}22\, \sigma_{bcu}$

O.K.



Member F

No force in member; just for stability.
Assume 150 x 150 mm



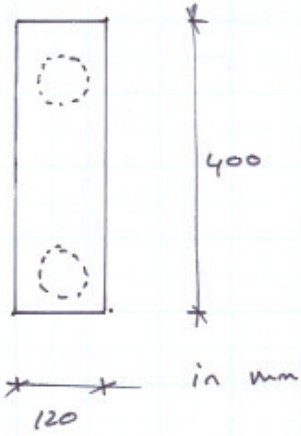
Member G

$F = 2400 \text{ kN}$
$\sigma_{c;p} = 67 \text{ MPa}$
$A_c = 36000 \text{ mm}^2$
$A_p = \frac{2400000}{1120} = 2150 \text{ mm}^2 = 15 \times \phi 15.7$ or $22 \times \phi 12.9$

2x tendons with 12 strands ϕ 12.9
VSL - VMA type EC: ϕ 72
$A_{tendon} = 4072 \text{ mm}^2$

$A = 36000 + 2 \cdot 4072 = 44150 \text{ mm}^2$

$= 400 \times 120 \text{ mm}$



CORSMIT
RAADGEVEND
INGENIEURSBUREAU
BV

Postbus 208
2280 AE Rijswijk (ZH)
tel. 070-394 93 05
fax 070-394 13 96

werk no. MF Afstnderen

berek. no. dimensionering staven

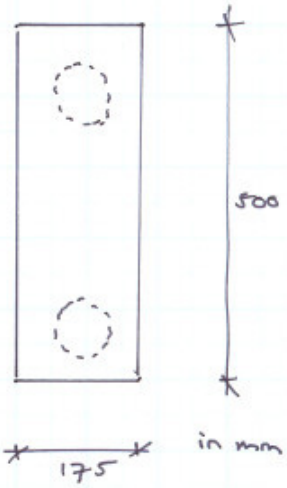
blz. no. 4/4

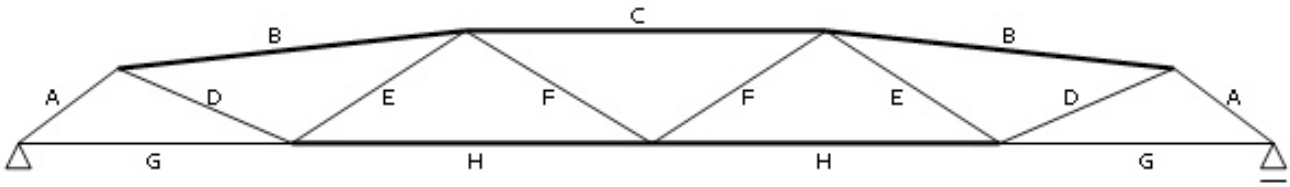
Member H

$F = 4800$ kN
$\sigma_c = 67$ MPa
$A_c = \frac{72000}{1120}$ mm^2
$A_p = \frac{4800000}{1120} = 4286$ mm^2
$= 29 \ \phi 15.7$ or $43 \ \phi 12.9$

2 x tendons with 22 strands ϕ 12.9
VSL-VMA type EC : ϕ 92
$A_{tendon} = 6650$ mm^2

$A = 72000 + 2 \cdot 6650 = 85300$ mm^2
$= 500 \times 175$ mm





Member lettering as used in the calculations

APPENDIX F CANDIDATE AND COMMITTEE DATA

Graduation candidate

Surname:	Flint
Given names:	Martin Gerard
Address:	Laan van Overvest 34 2613 DM Delft The Netherlands
Telephone:	+31 6 41 770 565
Email:	m.g.flint@student.tudelft.nl martinflint@hotmail.com
Date of birth:	1 August 1983
Nationality:	Dutch
Working location:	Corsmit Consulting Engineers Winston Churchill Tower, 21st floor Sir Winston Churchilllaan 366B 2285 SJ Rijswijk (ZH) The Netherlands +31 70 394 92 05
	Delft University of Technology Faculty of Civil Engineering and Geosciences Stevinweg 1, room 0.72 2628 CN Delft The Netherlands

Graduation committee

Prof.dipl.-ing. J.N.J.A. Vamberský
Chairman

Position:	Professor Building Structures
University:	Delft University of Technology
Faculty:	Civil Engineering and Geosciences
Department:	Buildings and Civil Engineering Structures
Address:	Stevinweg 1, room 1.36 Stevin II 2628 CN Delft The Netherlands
Telephone:	+31 15 278 54 88
Email:	J.N.J.A.Vambersky@tudelft.nl

Prof.dr.ir. J.G. Rots
Structural mechanics

Position:	Professor Structural Mechanics
University:	Delft University of Technology
Faculty:	Civil Engineering and Geosciences
Department:	Structural Mechanics
Address:	Stevinweg 1, room 6.71 2628 CN Delft The Netherlands
Telephone:	+31 15 278 37 99
Email:	J.G.Rots@tudelft.nl

Ir. J.L. Coenders
Computational design specialist, daily supervisor

Position:	Researcher
University:	Delft University of Technology
Faculty:	Civil Engineering and Geosciences
Department:	Structural Design Lab
Address:	Stevinweg 1, room 1.60 Stevin II 2628 CN Delft The Netherlands
Telephone:	+31 15 278 57 11
Email:	J.L.Coenders@tudelft.nl

Dr.-ing. S. Grünewald
Concrete technology specialist

Position:	Assistant professor
University:	Delft University of Technology
Faculty:	Civil Engineering and Geosciences
Department:	Design and Construction, Concrete Structures
Address:	Stevinweg 1, room 1.04 Stevin II 2628 CN Delft The Netherlands
Telephone:	+31 15 278 45 80
Email:	S.Grunewald@tudelft.nl

Ir. J.W. Welleman
Structural mechanics specialist

Position:	Senior lecturer
University:	Delft University of Technology
Faculty:	Civil Engineering and Geosciences
Department:	Structural Mechanics
Address:	Stevinweg 1, room 6.65 2628 CN Delft The Netherlands
Telephone:	+31 15 278 48 56
Email:	J.W.Welleman@tudelft.nl

Dr.ir. M.A.N. Hendriks
Computational mechanics specialist

Position:	Assistant professor
University:	Delft University of Technology
Faculty:	Civil Engineering and Geosciences
Department:	Structural Mechanics
Address:	Stevinweg 1, room 6.70 2628 CN Delft The Netherlands
Telephone:	+31 15 278 69 88
Email:	M.A.N.Hendriks@tudelft.nl

REFERENCES

Books

[1] AFGC/SETRA. 2002, *Ultra High Performance Fibre-Reinforced Concretes; Interim Recommendations*, Association Française de Génie Civil, France.

[2] Bentley, P.J. 1999, *Evolutionary Design by Computers*, Morgan Kaufmann Publishers, San Francisco, CA, USA.

[3] Burns, S.A. (ed). 2002, *Recent Advances in Optimal Structural Design*, American Society of Civil Engineerns, Reston, VA, USA.

[4] Camazine, S., Deneubourg, J.-L., Franks, N.R., Sneyd, J., Theraulaz, G. and Bonabeu, E. 2001, *Self-Organization in Biological Systems*, Princeton University Press, Princeton, NJ, USA.

[5] Coenders, J.L. (ed), 2006. *Structural Design – Special Structures*, Delft University of Technology, The Netherlands.

[6] CUR/NNI. 1995, *NEN 6720, TGB 1990 Regulations for concrete. Structural requirements and calculation methods*, Nederlands Normalisatie Instituut, Delft, The Netherlands.

[7] De Witte, F.C. (ed). 2007, DIANA – Finite Element Analysis User's Manual release 9.2, TNO DIANA, Delft, The Netherlands.

[8] Dorigo, M. and Stützle, T. 2004, *Ant Colony Optimization*, MIT Press, Cambridge, MA, USA.

[9] Feoktistov, V. 2006, *Differential Evolution: in Search of Solutions*, Springer New York, NY, USA.

[10] Hendriks, Ch.F. 1999, *Duurzamen Bouwmaterialen* (in Dutch), Æneas, Best, The Netherlands.

[11] Kelly, K. 1994, *Out of Control; The new biology of machines, social systems and the economic world*, Perseus Publishing, New York, NY, USA.

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

[13] Nawy, E.G. 1996, *Fundamentals of High Strength High Performance Concrete*, Longman group, Harlow, England.

[14] Oosterhoff, J. 1978, *Constructies; Momenten uit de geschiedenis van het overspannen en ondersteunen* (in Dutch), Delft University Press, The Netherlands.

[15] Owen, J.B.B. 1965, *The Analysis and Design of Light Structures,* Edward Arnold Publishers Ltd, London, UK.

[16] Peters, T. 1997, *Vitruvius; Handboek Bouwkunde*. Dutch translation of Vitruvius' Architectura, Em. Querido's Uitgeverij B.V., Amsterdam, The Netherlands.

[17] Walraven, J.C. and Galjaard, J.C. 1997, *Voorgespannen Beton* (in Dutch), ENCI Media, 's-Hertogenbosch, The Netherlands.

[18] Wagemans, L.A.G 2003, *Infomap nieuwe stijl* (in Dutch), Sectie Algemene Constructieleer, TU Delft, The Netherlands

[19] Xie, Y.M. and Steven, G.P. 1997, *Evolutionary Structural Optimization*, Springer London, UK.

Scientific articles

[20] Acker, P. and Behloul, M. 2004, 'Ductal® Technology: a Large Spectrum of Properties, a Wide Range of Applications', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 11-24.

[21] Behloul, M. and Lee, K.C. 2003, 'Ductal® Seonyu Footbridge', *Structural Concrete*, vol. 4, no. 4, pp. 195-201.

[22] Bornemann, R. and Faber, S. 2004, 'UHPC with steel- and non-corroding high-strength polymer fibres under static and cyclic loading', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 673-682.

[23] Camp, C.V. and Bichon, B.J. 2004, 'Design of Space Trusses using Ant Colony Optimization', *Journal of Structural Engineering*, May 2004, pp. 741-751.

[24] Camp, C.V., Bichon, B.J. and Stovall, S.P. 2005, 'Design of Steel Frames using Ant Colony Optimization', *Journal of Structural Engineering*, March 2005, pp. 369-379.

[25] Deb, K. and Gulati, S. 2001, 'Design of Truss-Structures for Minimum Weight using Genetic Algorithms', *Finite Elements in Analysis and Design*, vol. 37, no. 5, pp. 447-465.

[26] Deneubourg, J.-L., Aron, S., Goss, S. and Pasteels, J.M.

1990, 'The Self-Organizing Exploratory Pattern of the Argentine Ant', *Journal of Insect Behavior*, vol. 3, no. 2, pp. 159-168.

[27] Dorigo, M., Maniezzo, V. and Colorni, A. 1996, 'Ant System: Optimization by a Colony of Cooperating Agents', *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41.

[28] Droll, K. 2004, 'Influence of additions on ultra high performance concretes – grain size optimization', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 285-302.

[29] Geisenhanslüke, C. and Schmidt, M. 2004, 'Methods for Modelling and Calculation of High Density Packing for Cement and Fillers in UHPC', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 303-312.

[30] Goss, S., Aron, S., Deneubourg, J.-L. and Pasteels, J.M. 1989, 'Self-organized Shortcuts in the Argentine Ant', *Naturwissenschaften*, vol. 76, pp. 579-581.

[31] Hajar, Z., Lecointre, D., Simon, A. and Petitjean, J. 2004, 'Design and Construction of the world first Ultra-High Performance Concrete road bridges', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 39-38.

[32] Hajar, Z., Simon, A., Lecointre, D. and Petitjean, J. 2003, 'Construction of the first Road Bridges made of Ultra-High Performance Concrete', in *Proceedings of the 3rd International Symposium on High Performance Concrete*. Precast/Prestressed Concrete Institute, Orlando, FL, USA.

[33] Hegger, J., Tuchlinski, D. and Kommer, B. 2004, 'Bond Anchorage Behavior and Shear Capacity of Ultra High Performance Concrete Beams', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 351-360.

[34] Heinz, D., Dehn, F. and Urbonas, L. 2004, 'Fire Resistance of Ultra High Performance Concrete (UHPC) – Testing of Laboratory Samples and Columns under Load', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 703-716.

[35] Heinz, D. and Ludwig, H.M. 2004, 'Heat Treatment and the Risk of DEF Delayed Ettringite Formation in UHPC', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 717-730.

[36] Huang, X., Xie, Y.M. and Burry, M.C. 2006, 'A new Algorithm for Bi-directional Evolutionary Structural Optimization', *JSME International Journal*, series C, vol. 49, no. 4, pp. 1091-1099.

[37] Karlsen, J., CRC – a Description. Retrieved 26 November 2007, from *www.crc-tech.com: documentation*.

[38] Kepler, J. 2002, 'Structural optimization as a design and styling tool'. Retrieved 17 December 2007, from *www.caam.rice.edu/~caam210/fibnet/kepler.pdf*.

[39] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. 1983, 'Optimization by Simulated Annealing', *Science*, vol. 220, no. 4598, pp. 671-680.

[40] Korpa, A. and Trettin, R. 2004, 'The use of synthetic colloidal silica dispersions for making HPC and UHPC systems, preliminary comparison results between colloidal silica dispersions and silica fumes (SF)', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 155-164.

[41] Lee, K.Y. 'Frei Otto, Bodo Rasch: Finding Form'. Retrieved 6 December 2007, from *http://www.yeul.net/Finding%20Form.pdf*.

[42] Luh, G.-C., Wu, C.-Y. and Lin, C.-Y. 2005, 'Multi-modal topological optimization of structure using ACO algorithms', in *Proceedings of the 8th Joint Conference on Information Sciences (JCIS 2005)*, Salt Lake City, UT, USA, pp 479-482.

[43] Ma, J. and Dietz, J. 2002, 'Ultra High Performance Self Compacting Concrete', *Leipzig Annual Civil Engineering Report*, 2002, no. 7, pp 33-42.

[44] Ma, J., Orgass, M., Dehn, F., Schmidt, D. and Tue, N.V. 2004, 'Comparative Investigations on Ultra-High Performance Concrete with and without Coarse Aggregates', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 205-212.

[45] Mitsui, K. 2006, 'Optimal Shape and Topology of Structure Searched by Ants' Foraging Behavior', *Report of the Research Institute of Industrial Technology, Nihon University*, no. 83, pp. 1-11.

[46] Orgass, M. and Klug, Y. 2004, 'Fibre Reinforced Ultra-High Strength Concretes', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 637-648.

[47] Perry, V.H. and Zakariasen, D. 2004, 'First use of Ultra-High Performance Concrete for an innovative Train Station Canopy', *Concrete Technology Today*, August 2004, pp. 1-2.

[48] Resplendino, J. and Petitjean, J. 2003, 'Ultra-High Performance Concrete; first Recommendations and Examples of Application', in *Proceedings of the 3rd International symposium on High Performance Concrete*. Precast/Prestressed Concrete Institute, Orlando, FL, USA.

[49] Resplendino, J. 2006, 'Les Bétons Fibrés Ultra Performants' (Presentation in French), *Colloque sur les Ouvrages d'Art*, Toulouse, France.

[50] Richard, P. and Cheyrezy, M. 1995, 'Composition of Reactive Powder Concretes', *Cement and Concrete Research*, vol. 25, no. 7, pp. 1501-1511.

[51] Samdani, S.A., Belambe, V. and Patil, A. 2004, 'Discrete Optimization of Trusses using Ant Colony Metaphor', *International conference on world of innovations in structural engineering*, 2004.

[52] Serra, M. and Venini, P. 2005, 'On some Applications of Ant Colony Optimization Metaheuristic to Structural Optimization Problems', *6th World Congress of Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil.

[53] Servant, C. 2006, 'l'Utilisation de Nouveaux Matériaux

pour la Construction du Viaduc et de la Gare de Péage de Millau' (Presentation in French), *Colloque sur les Ouvrages d'Art*, Toulouse, France.
[54] Shim, P. and Manoochehri, S. 1997, 'Generating Optimal Configurations in Structural Design using Simulated Annealing', *International Journal for Numerical Methods in Engineering*, vol. 40, no. 6, pp. 1053-1069.
[55] Stützle, T. and Hoos, H.H. 2000, 'MAX-MIN Ant System', *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889-914.
[56] Talebinejad, I., Bassam, S.A., Iranmansesh, A. and Shekarchizadeh, M. 2004, 'Optimizing mix proportions of Normal Weight Reactive Powder Concrete with Strengths of 200-350 MPa', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 133-142.
[57] Tanaka, Y., Musya, H., Ootake, A., Shimoyama, Y. and Kaneko, O. 2002, 'Design and Construction of Sakata-Mirai Footbridge using Reactive Powder Concrete', in *Proceedings of the 1st fib Congress, Concrete Structures in the 21st Century,* Japan Concrete Institute, Japan.
[58] Tang, M.-C. 2004, 'High Performance Concrete – Past, Present and Future', in M. Schmidt, E. Fehling and C. Geisenhanslüke (eds), *Ultra High Performance Concrete (UHPC)*, Kassel University Press GmbH, Germany, pp. 3-10.
[59] Vicenzino, E., Culham, G., Perry, V.H., Zakariasen, D. and Chow, T.S. 2005, 'First use of UHPFRC in thin Precast Concrete Roof Shell for Canadian LRT Station', *PCI Journal*, September-October 2005, pp. 50-67.
[60] Walraven, J.C. 2006, 'Ultra-hogesterktebeton: een materiaal in ontwikkeling' (in Dutch), *Cement*, 2006, no. 5, pp 57-61.
[61] Weiler, B., and Grosse, C. 1996, 'Pullout Behaviour of Fibers in Steel Fibre Reinforced Concrete', *Otto Graf Journal*, 1996, no. 7, pp. 116-127.

Theses

[62] Grünewald, S. 2004, *Performance-based Design of Self-Compacting Fibre Reinforced Concrete*, PhD thesis, Delft University of Technology, The Netherlands.
[63] Lappa, E.S. 2007, *High Strength Fibre Reinforced Concrete; Static and Fatigue behaviour in Bending*, PhD thesis, Delft University of Technology, The Netherlands.
[64] Veenendaal, D. 2008, *Evolutionary optimization of fabric-formed structural elements*, MSc thesis, Delft University of Technology, The Netherlands.
[65] Voorde, A. ten. 2004, *Kokerligger in zeer hogesterktebeton* (in Dutch), MSc thesis, Delft University of Technology, The Netherlands.
[66] Vries, T. de. 2008, *Modulaire Fiets- en Voetgangersbrug in Vezelversterkt Hoge- en Zeer Hogesterktebeton* (in Dutch), MSc thesis, Delft University of Technology, The Netherlands.

Technical sheets

[67] Eiffage. 2004, 'Caractéristiques Mécaniques et Physiques' (in French). Retrieved 26 November 2007, from *www.bsieiffage.com*.
[68] Lafarge FR. 2007, 'Ductal®-AF fiche de Caractéristiques Techniques' (in French). Retrieved 26 November 2007, from *www.ductal-lafarge.com*.
[69] Lafarge NA. 2006, 'Technical Characteristics of Ductal® with Metallic Fibres (imperial)'. Retrieved 26 November 2007, from *www.imagineductal.com*.
[70] Vicat. 2004, 'Fiche technique BCV Fibres Métalliques' (in French). Retrieved 27 November 2007, from *www.brutdebeton.com*.

Websites

[71] http://en.wikipedia.org/wiki/Local_minimum.
[72] http://en.wikipedia.org/wiki/Lagrange_multiplier.
[73] http://www-math.mit.edu/~djk/18_022/chapter04/section03.html.
[74] http://en.structurae.de.