Structural Optimization of Free-Form Grid Shells

Milos DIMCIC
PhD student
Institute of Building Structures and Structural Design (itke), University of Stuttgart, Germany
m.dimcic@itke.uni-stuttgart.de

Milos Dimcic, born 1982, began his PhD studies at the Stuttgart University in 2009, after receiving his degree in Architecture from Belgrade University. He worked for Knippers Helbig office from 2007-09 on various large scale projects.

Jan KNIPPERS
Professor
Institute of Building Structures and Structural Design (itke), University of Stuttgart, Germany
info@itke.uni-stuttgart.de

Jan Knippers, born 1962, Dipl.Ing. in Civil Engineering 1989 and Dr. Ing. in 1992 at the Technical Univ. of Berlin. Partner in Knippers Helbig Advanced Engineering. Professor at University of Stuttgart.

Summary

In the 21st century, as free-form design gains popularity, free-form grid shells are becoming a universal structural solution, enabling merger of structure and façade into a single layer – *a skin* [1]. This paper shows some of the results of a research project concerned with the optimization of grid structures over some predefined free form shape, with the goal of generating a stable and statically efficient structure. It shows how combining design and FEM software in an iterative, Genetic Algorithms based, optimization process, stress and displacements in grid shell structures can be significantly reduced, whereby material can be saved and stability enhanced.

Keywords: Free-form, Grid shell, Genetic Algorithms, Structural optimization, Voronoi diagram

1. Introduction

At the end of the 20th century we witnessed the appearance of the first steel free-form grid shell structures with all structural members different (unique), since there was no longer any substantial difference in cost between producing 1000 unique objects and 1000 identical ones [1]. In the 21st century the field of free-form grid shell structural design is being further developed, but structural design and optimization techniques are still mostly based on the trial and error approach. This is usually not sufficient, since, when dealing with free-form shapes, experience based intuition rarely succeeds in finding the optimal solution, due to the high geometrical complexity and a huge search space, i.e., large number of possible solutions. In this paper, some of the results of the comprehensive research dealing with the automation of this optimization process will be presented. In order to not limit the creativity of architects, the idea was to generate the best structural solution over some already defined shape. Instead of form-finding we are trying to find the best geometry and topology of a grid shell, while keeping it on the specific surface all the time. The proposed method of structural optimization is constructed as a C++ based plug-in for Rhinoceros 3D, one of the main NURBS (**N**on **U**niform **R**ational **B**-**S**plines) geometry based modelling tools used by architects for free-form design today. The algorithm communicates iteratively with FEM software for static analysis. In this case Oasys GSA commercial FEM software is used.

2. From NURBS to Grid

Before explaining the structure of the algorithm, the method of automatic grid generation over a given free-form NURBS surface has to be addressed. For this purpose, and within the presented research, the decision was made to use Voronoi Diagrams (Figure 1) [2], for two main reasons. First,

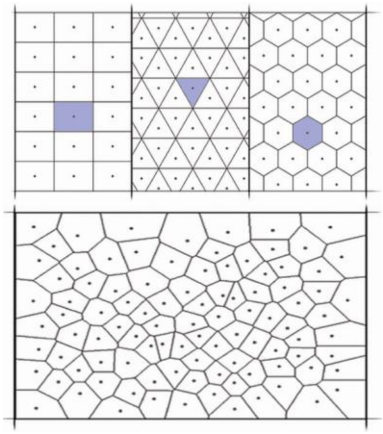


Fig. 1: Voronoi diagram in 2D

NURBS surfaces are mathematically represented over two parameters (*uv*) and algorithms for Voronoi diagram generation in 2D (in plane) can be therefore mapped onto the surface, using a direct *xy-uv* transformation. Second, depending on the disposition of the Voronoi points, a large number of different, natural structures can be generated, but also structures with a regular grid pattern (like triangular, quadrangular and hexagonal), as depicted in Figure 1. *Non-uniform* Voronoi structures (Figure 1-down) however have polygons with very different edge lengths and angles which are usually not acceptable for the purpose of grid shell design. In order to solve this problem, the Force-Density Method [3,4] was used and adapted (expanded) to *relax* a grid, while keeping it on the predefined surface. This resulted in a new type of structure, named Voronax (Voronoi + Relax), which is created by relaxing a Voronoi structure generated over a given free-form surface (Figure 2). Voronax is a foam-like structure since the principle of its generation is a dynamic search of equilibrium, similar to the process happening within actual foam. Its polygons have much more similar angles and lengths, and the automatic generation of a Voronax structure over some predefined shape is used as a basis for the presented structural optimization method. By controlling the distribution of Voronoi points over the surface, we directly influence the generated grid shell. Therefore, letting the points be the main variables of the optimization process we can apply Genetic Algorithms in order to find their best distribution, i.e., a disposition of points used to generate the most efficient grid structure according to the defined criterion.

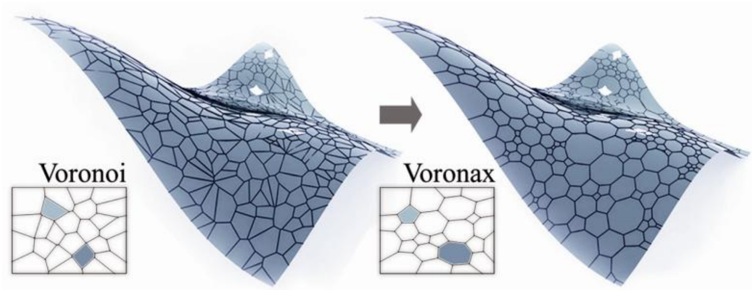


Fig. 2: Relaxation of a Voronoi structure = Voronax grid

3. Optimization Algorithm – Genetic Algorithms

Genetic Algorithms (GAs) are chosen as a suitable method for multi-objective and highly non-linear optimization. It is a stochastic method, based on the principle of evolution, within which a random population of individuals is generated (grid shells in our case) at the beginning. The best individuals, according to their *fitness*, are then chosen for reproduction and with specific *crossing* techniques, solutions are combined to bring new offspring and in that way form a new generation. The crossing methods ensure the heritage of good genes, thus enabling the whole process to converge toward the best fitness solution. Specific *mutation* algorithms enable random alteration of individuals in order to introduce diversity and ensure the better exploration of the search space, thus avoiding convergence to local optima. This loop (Figure 3) then continues until the *satisfactory solution* is found. In our case, we are searching for a grid shell structure with minimum material usage (minimum weight) and minimum potential energy of the system, the same thing that Nature does with its own structures through millions of years of evolution. Grid shells can be evaluated optically or statically, according to the defined fitness function, and in this paper the focus is on the statical optimization. More on the basics of the Genetic Algorithms application can be found in [4].

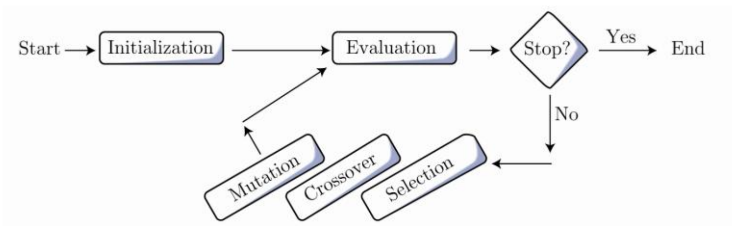


Fig. 3: Basic GAs Loop

3.1 Input Parameters

The goal of this research from the beginning was to make a universal method for grid shell optimization, adaptable, easily expandable and with a large number of variables, i.e., with a clear definition of boundaries and settings within which we expect our solution to be generated. Therefore a plug-in was developed so that the user can: **1.** Choose the surface over which the grid will be generated, **2.** Chose the basic pattern of the grid (e.g. Delaunay triangulation [2], quadrangular, Voronoi, Voronax, etc.), **3.** Set a support combination (e.g. all four edges, two edges, fully restrained, movable, etc.) **4.** Set a load combination (any load combination definable in FEM software), **5.** Set material properties, **6.** Set cross-section of the structural members, 7. Define the fitness function (e.g. minimize Von Mises stress, minimize displacement, maximize Load buckling factor, etc.), **8.** Define one or more penalty functions (e.g. limit the length of a member, limit the size of a polygon, limit the stress generated in one member, etc.), **9.** Set GAs parameters (e.g. crossover and mutation probability, number of individuals, number of generations, etc.). Each one of these settings (Figure 4) can be easily expanded and redefined. After they are chosen, the optimization process begins and the algorithm converges toward the best solution for that combination of input settings, whatever they are.

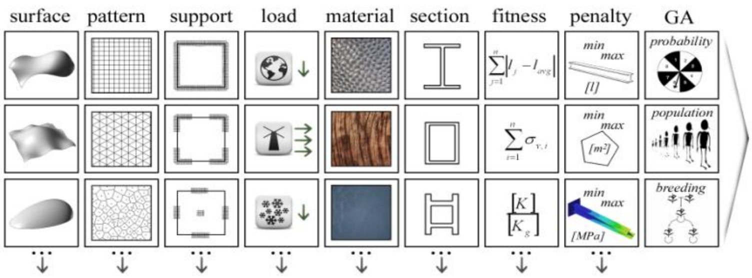


Fig. 4: Input parameters, expandable and changeable

3.2 Basic Algorithmic Loop

Genetic Algorithms work with a *chromosome* representation. In this research the chromosome is formed as a string of real-valued numbers which are later on transformed into the *uv* coordinates on the surface. This is done with a specific set of *decoding* functions. The *uv* coordinates are used to generate points from which a Voronoi diagram (over a given surface) is calculated and eventually relaxed, resulting in a Voronax grid structure. Each grid shell in the algorithm goes through an eleven step process depicted in Figure 5. First, the basic GAs operations (selection, crossing, mutation) are performed, followed by the *decoding* part (or *generation*) where the chromosome is transformed into a grid shell and prepared for FEM static analysis. Step 8 refers to an automatic call of the FEM software where the static analysis of the generated grid shell is performed. When the needed results are obtained (e.g. forces, moments, displacements, etc.) the evaluation according to the chosen fitness function is carried out, and the solution is *penalized* if it violates any of the specified constraints. The fitness value and the violation of constraints are then combined and *scaled* into one final fitness value of the generated individual solution. In a usual optimization there are 50 grid shells in a generation, and the process lasts for 400-700 generations, thus sometimes generating more than 30.000 solutions. All the solutions are kept in specific text files that enable their recreation, i.e., extraction and drawing of any of the generated grid shells in the process.

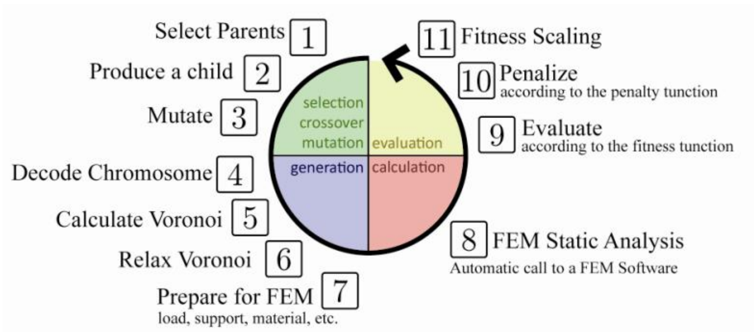


Fig. 5: Basic Algorithmic Loop for one grid shell solution

4. Optimization

One experiment will be shown here, done with the Voronax pattern, and one particular method of its application will be presented. Although different patterns can be selected at the beginning, here it will be demonstrated how the Voronax pattern can be used to determine an optimal disposition of the grid density in order to generate a statically more efficient structure. The surface (a vertical *wall*) used in the experiment is shown in Figure 6, as well as the input settings used for the optimization. All joints generated on the edges of the surface are set to be restrained from movement and rotation in all directions. Load is applied as self-weight of the structural members and horizontal surface load of 1 KN/m^2. The horizontal load is applied by calculating the surface of each generated cell (polygon), dividing it and transferring the load to the structural joints, as shown down in Figure 6. The fitness function used is the minimization of Von Mises stress (σ_v). For each structural member in the grid shell the simplified version of Von Mises stress (Eq. 1,2,3,4) is calculated at both of its ends (denoted as 0 and 1). Those values are summed up for all (n) structural members resulting in a fitness value ($F(x)$) for the entire structure, which we try to minimize (Eq. 5).

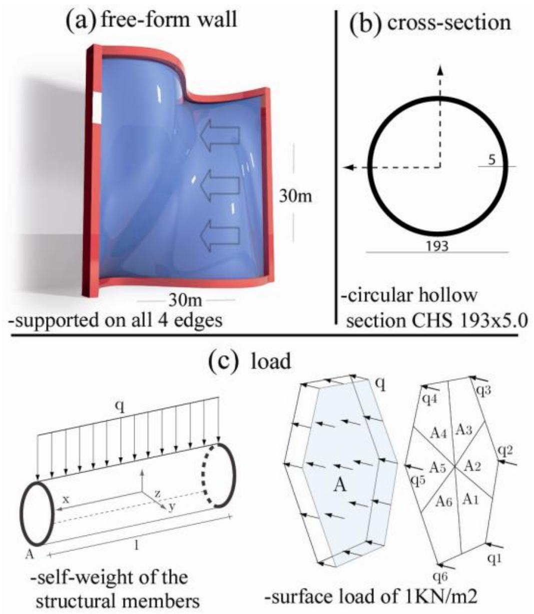


Fig. 6: Surface, Cross-section and Load used in the experiment

$$\sigma_v = \sqrt{\sigma_{xx}^2 + 3\tau_{xy}^2 + 3\tau_{xz}^2} \tag{1}$$

$$\sigma_{xx} = \frac{F_x}{A} \pm \frac{M_y}{W_y} \pm \frac{M_z}{W_z} \tag{2}$$

$$\tau_{xy} = \frac{F_y}{A_y} \tag{3}$$

$$\tau_{xz} = \frac{F_z}{A_z} \tag{4}$$

Minimize: $$F(x) = \sum_{i=1}^{n} [\sigma_{v,i,0} + \sigma_{v,i,1}] \tag{5}$$

For the presented experiment, structural members with a circular hollow cross-section (CHS 193x5.0) were used. All generated members have the same section, since the goal is to find the best topology and geometry of the grid, i.e., to minimize stress by keeping the mass relatively the same. Within the research, experiments were done with properly oriented rectangular cross-section and with proper wind load (normal to the surface at all points). An optimization with these settings however introduces a different set of problems which are not the focus of this paper, and that is why, for the presented optimization, the settings were simplified using a circular section and horizontal load. This however has no effect on the efficiency of the optimization process, since, as mentioned before, it works for any kind of input parameter combination.

4.1 Voronax optimization

The Voronax pattern optimization is performed with a 150 point chromosome. That means that for each individual solution, 150 points are generated over a surface, turned into a Voronoi diagram, which is then relaxed resulting in a Voronax grid structure. In Figure 7, there are two graphs showing the convergence of the optimization process after 550 generations (27.500 generated individual grid shell solutions). The graph on the top shows the progress of the average fitness value in each generation (calculated from 50 individuals). The graph underneath shows fitness values of the best individual solution (grid shell) in each generation. It can be seen how both graphs show a constant descent of the total Von Mises stress generated in the structure and a steady convergence. Here we also introduce a displacement factor. Namely, for each joint in the structure its

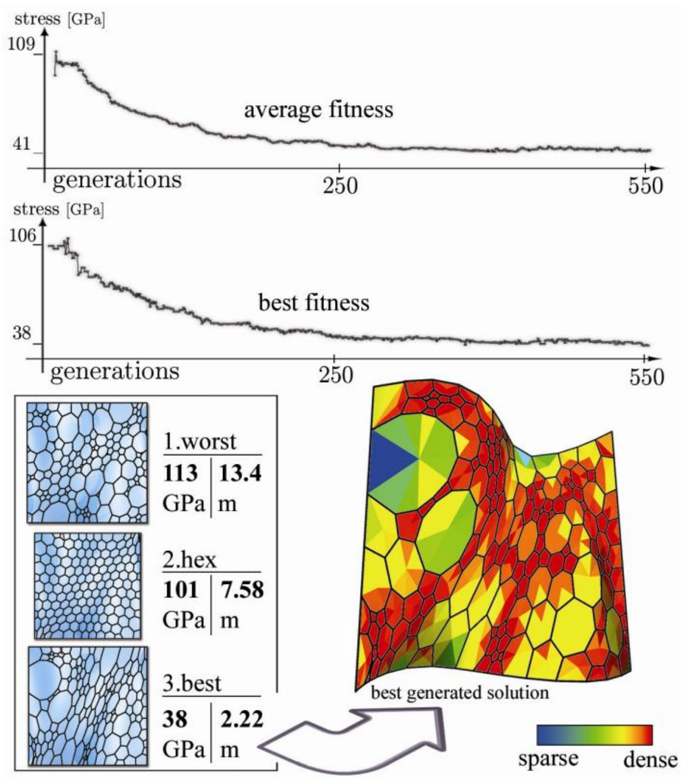


Fig. 7: Results of the optimization process performed with the Voronax pattern

displacement (movement) is calculated (d_i) as a vector in space, derived from the movements in all three (x,y,z) directions (Eq. 6). The magnitude of all joint movements is then summed up, resulting in a *total displacement* of the structure (Eq. 7).

$$d_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \qquad (6)$$

$$F(x) = \sum_{i=1}^{n} d_i \qquad (7)$$

In Figure 7, down and on the left, depicted from the front view, there is:

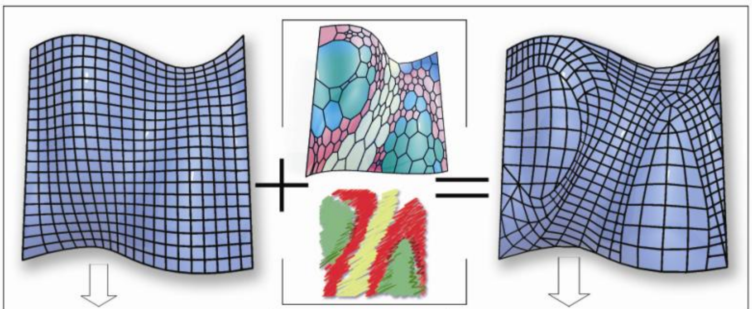
1. The worst generated solution, created randomly in one of the first generations, having 113 GPa as the total amount of Von Mises stress and 13.4*m* of total joint displacement.

2. For comparison, a hexagonal structure is used, representing basically a uniform version of the Voronax grid. The reason for this is that Voronax keeps the topology of the Voronoi structure after relaxation, which means that on average its polygons have ~ 6 edges [5] and the joints have a 3-member connection (as in a hexagonal grid). This uniformly distributed grid only shows a slightly better performance (101 GPa and 7.58*m*) than the worst generated solution.

3. The best generated solution from one of the latest generations has the smallest amount of Von Mises stress generated in its members (38 GPa), i.e., three times smaller than the worst generated solution and 6 times smaller amount of displacement (2.22*m*). In Figure 7, on the right-hand side, there is a colour analysis of this Voronax grid solution, showing the distribution of the grid density (from blue=sparse to red=dense).

There is a number of different ways of how this information can be used in a grid shell design. Following the *advice* of the GAs algorithm we can use different techniques, from controlled relaxation to the combination of different patterns to achieve a statically efficient design. One of those possibilities will be presented now.

4.2 Interpretation and Application



840	structural members	830	
441	joints	434	
51 GPa	∑ Von Mises stress	**44** GPa	-13 %
2.44 m	∑ Deformation	**1.84** m	-25 %

Fig. 8: Redesign based on the GAs optimization results

We can generate a uniform quadrangular structure over our free-form wall as shown on the left-hand side in Figure 8. Then we can try to interpret the *intention* of the GAs optimization process. It can be seen that the best offered structural solution has enlarged grid density around the convex parts (red area in two representations in the middle of Figure 8), thus stiffening them up, and *stretched* the cells over the diagonal between the two convex parts (yellow area). Using this information we can try to generate a quadrangular structure with similar number of joints and members, as depicted on the right-hand side of the Figure. Doing so, we get a quadrangular structure with 13% less generated stress and 25% smaller amount of displacement. By combining different patterns (triangular, quadrangular, hexagonal) we can develop different solutions, knowing the distribution of grid density (hence stiffness) that produces optimal results according to the desired criteria.

5. Conclusion

This paper presents an automated method of grid shell optimization that offers optimal structural solutions over some given free-form surface. The focus is on the fact that no *approximation* or pure trial and error method has to be involved in the structural design process if we use the proposed optimization method. The main advantage of the Voronax structure is that it can be easily *interpreted* most of the time. For example, in Figure 9, there are results of the optimization done over two flat vertical surfaces, with the same load combination applied as in the examples above (self-weight of the structural members + horizontal load). The joints are restrained on four corners of the surface in the structure on the upper part of the figure, and in the middle of the surface edges on the structure depicted down (restrained areas are marked red). For each option the best solution obtained in an optimization process can be seen, and next to it a *look through* the last generated generation is depicted. Namely, if we take all 50 solutions of one generation and line them up one behind the other, we can get a comprehensive picture of the *intention* of the optimization process. (It can be seen how the centre part in both cases has larger cells, stabilized with the *O*-shaped formation of denser cells in the upper case and the *X*-shape formation in the case depicted underneath.)

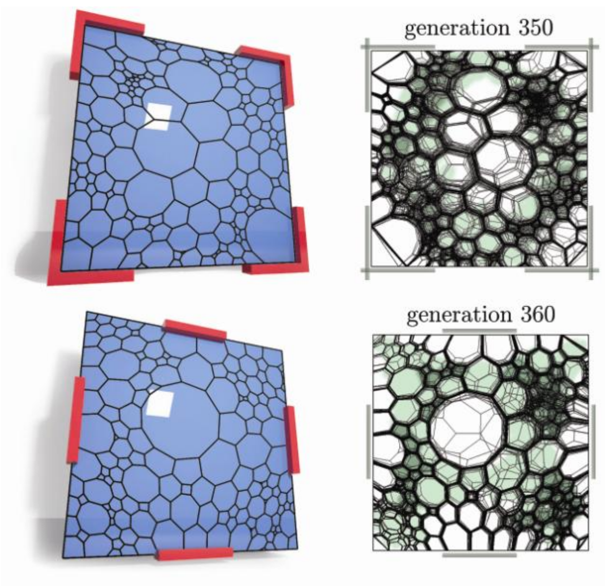


Fig. 9: Different support combinations

These experiments are a part of the comprehensive research done with different shapes, fitness functions, penalty functions, support and load combinations and different patterns. Optimizations are done not only as single-objective but also as multi-objective ones, showing that, depending on the free-form shape and the grid pattern, we can generate grid shells that have up to 6 times less Von Mises stress and up to 10 times less displacement when compared to a regular (uniform) structure, generated with the same number of structural members and over the same given surface.

[1] KOLAREVIC B., Architecture in Digital Age – Design and Manufacturing. Taylor and Francis, 2005.

[2] DE BERG M., CHEONG O., VAN KREVELD M., OVERMARS M., *Computational Geometry*, Springer Verlag, 1997.

[3] GRÜNDIG L., MONCRIEFF E., SINGER P., STRÖBEL, A History of the Principal Developments and Applications of the Force Density Method in Germany 1970-1999, in IASS-IACM, Chania-Crete, 2000.

[4] Linkwitz, K. and Schek, H.-J., Einige Bemerkungen zur Berechnung vonvorgespannten Seilnetzkonstruktionen', Ingenieur-Archiv 40, 145-158, 1971

[5] GOLDBERG D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley , 1989

[6] SACK J.R., URRUTIA J., *Handbook of Computational Geometry*, North Holland, 1999