



Computational approach for form-finding optimal design

Saaranya Kumar Dasari¹ · Nicholas Fantuzzi² · Patrizia Trovalusci¹ · Roberto Panei³

Received: 25 January 2022 / Accepted: 10 November 2022
© The Author(s) 2022

Abstract

In this paper, an optimization strategy for a canopy, based on computational modelling approaches is presented. The design approach is applied to a realistic roof structure of an ecological island (waste collection centre) and has been completely redesigned with the aid of a Genetic Algorithm and a Dynamic Relaxation Algorithm. The preliminary design of the roof structure can be formulated as a shape optimization problem, involving functional needs and constraints at different scales of the structure. The proposed hypothesis solution was studied by using an optimization procedure through algorithms in the software Rhinoceros3D®/Grasshopper®. The main aim of this work is to explore different modelling approaches for form-finding that can be built from the use of numerical simulations based on algorithms. To this aim, the need to meet various requirements (structural, functional, formal) involving a team of architects and engineers can be interpreted as a matter of structural optimization.

Keywords Conceptual structural design · Structural optimization · Genetic algorithms · Dynamic relaxation · Finite element analysis

Introduction

In this study, we conceived a roof structure as a free form shell aimed at optimizing the shape using a Genetic Algorithm (GA) and a Dynamic Relaxation (DR) Algorithm. To this aim, fully parametric models are created in Rhinoceros3D® with aid of NURBS representation and interfaced with an in-house genetic algorithm and relaxation algorithm in Grasshopper® and Finite Element (FE) models are created for each design approach, in order to evaluate the structural performance under a given load condition. Results are discussed referring to the structural engineer and designer

who together must find the best architectural solution in the early stage of the design process. The emerging research field of conceptual structural design aims to fill the gaps between existing computational approaches and allow the real integration of structural inputs in the conceptual design process [1]. Computational techniques and attendant formal-spatial repertoire is maturing, allowing us to build unprecedented levels of spatial and morphological complexity [2]. Recent discourse on computational methods of form-finding linked to benefits of bio-inspired forms, their advocates list the potential for structural benefits derived from redundancy and differentiation as well as the ability to maintain multiple functions simultaneously.

Historically architects and engineers have persistently applied form-finding/form-improving [3] techniques that identify the process of designing optimal structural shapes by using experimental tools and strategies (physical models) to simulate a specific (expected) mechanical behavior. The end of the 20th century and the beginning of the 21st century are extraordinary innovations and developments in design and construction technology, which created a situation of architectural freedom, which can be referred to as any shape imaginable. In addition, the ever-increasing knowledge and control of calculations make computers a powerful design assistant that can analyze, calculate, and

✉ Saaranya Kumar Dasari
saaranyakumar.dasari@uniroma1.it

Nicholas Fantuzzi
nicholas.fantuzzi@unibo.it

Patrizia Trovalusci
patrizia.trovalusci@uniroma1.it

Roberto Panei
roberto.panei@uniroma1.it

¹ Sapienza University of Rome, Rome, Italy

² University of Bologna, Bologna, Italy

³ Municipal Urban Hygiene Company (AMA), Rome, Italy

geometrically control extraordinarily complex shapes. However, this huge technical development contrasts with the fact that in this period new materials or structural systems have not appeared with the relevance of the existing ones, which could suggest new shapes or typologies. Therefore, although the development of new forms is related to the appearance of new materials in other times, the origin of fractured, distorted, and formless shapes in contemporary architecture has nothing to do with new materials or systems, but with the technical, structural, and constructive abilities of nowadays. Due to the significant development of computer-aided design and virtual modeling technology, allowing more complex and precise analysis and calculations, physical models have gradually been replaced in recent decades. During the last decades, many computational methods have gained success in many fields of engineering and a wide range of possibilities are offered by sophisticated software. For this reason, FE methods are now essential tools for contemporary design. In fact, the continuing demand for lightweight, efficient and low-cost structures has a consequence the interest in structural optimization is increasing. The benefits of structural optimization and integrating numerical simulations into a conceptual design can create novel structural systems that are structurally more efficient.

The structural optimization of complex structures, like membranes, shells, and grid shells, has been historically an important research field since the debate on the equilibrium of arches and vaults of the 18th century [4] and the works of Antoni Gaudi, Pier Luigi Nervi, Sergio Musmeci, Heinz Isler, or Frei Otto and the experiences on physical models [5–8]. Computationally developed techniques like Evolutionary Structural Optimization (ESO) seems to approach the design problem with particular attention to architectural, and formal, aspects and replace the physical models of other times by computational analysis, applying these techniques as design tools, to establish a logic in the relation between the architectural shape and its structural support in contemporary architecture. The ESO topology optimization method is based on the simple concept that by systematically removing inefficient materials from the structure, the residual shape evolves toward an optimum [9]. Finally, in implementing our solutions, we are looking for insights on how general frameworks for optimization using algorithms could be realized.

In order to design a computational workflow, we needed the integration of different digital tools: a Computer-Aided Design (CAD) application, which can provide parametric control on shapes; an FE (Finite Element) solver; and an optimization algorithm. For this reason, the workflow was designed in the Rhinoceros3D®/Grasshopper® environment. Rhinoceros3D® (also called Rhino, or Rhino3D) is CAD software developed by Robert McNeel and Associates. The geometry in Rhino is based on NURBS curves

and surfaces, which is very efficient to generate free form shapes. Rhino has an inbuilt programmable plugin called Grasshopper® developed by David Rutten at Robert McNeel and Associates, which uses the graphic engine to display the outputs and allows an intuitive approach to parametric design and algorithmic modeling, without necessarily having to have advanced knowledge of scripting or programming [10, 11]. Grasshopper is a visual programming language that enables us to design by means of generative algorithms which use associated modeling and generative modeling [12]. Simulations were done using evolutionary solvers (genetic algorithm) of Galapagos and dynamic relaxation solvers of Kangaroo Physics (plugin) for form-finding, and Karamba 3D(plugin) for structural analysis in Grasshopper and Dlubal RFEM for FE analysis.

Structural optimization

For decades, structural forms have been designed through form-finding techniques by means of reverse hanging, minimal surfaces, and geometric forms, where they have strategies and specific methods for designing these forms. In the case of free form, the structure is designed through an optimization technique through the gesture of an idea, which is used to transform that free form shape, free because it was designed freely from structural principles and understanding material properties. Contemporary architecture practices have enabled a clear distinction between the role of the architect and the role of the engineer. In this paradigm, the architect is concerned with the shape and functionality of the building while the engineer is concerned with mathematical “firmitas”. Technological developments and contemporaneous architectonic research have tried to reintegrate the two fields by nesting shape and statics, as well as creativity and structural calculation [5–8].

Structural optimization can be defined as the process of finding the optimal conditions that give the maximum or minimum value of an objective function $f(\mathbf{x})$ subject to constraints in its own variables \mathbf{x} (also called parameters), the parameters that can influence, to optimize the structures, are called design variables [13]. The design variables could commonly be cross-sectional dimensions or member sizes, geometrical parameters, material properties, nodal positions depending on the optimization problem. In the field of structural engineering, the objective function can be represented as weight, volume, displacements, stresses, or cost of the structure, such that a minimal-material structure can be found [13]. The selection of an objective function can be one of the most important decisions in the whole optimization process and the choice of design variables must be done carefully as it influences the success of optimization. In the mathematical formulation of an optimization

problem, the values of design variables are subject to constraints. The restrictions that must be satisfied to produce a feasible design are called constraints [13]. Thus, two kinds of constraints can be identified: (1) inequality constraints which impose upper or lower limits to design variables and (2) equality constraints that impose that a design variable must assume a known value. An optimization problem could be expressed mathematically as follows:

$$\begin{aligned} \text{Objective function : } & \text{minimize } f(x) \\ \text{Subject to : } & g_j(x) \leq 0 \quad (j = 1, 2, \dots, n_g) \\ & h_j(x) = 0 \quad (j = 1, 2, \dots, n_h) \end{aligned}$$

where \mathbf{x} is the vector of design variables while $g_j(\mathbf{x})$ and $h_j(\mathbf{x})$ are respectively the inequality and equality constraints.

Depending on the optimization problem various optimization algorithms are available from classical methods to modern optimization methods. Most of them can be grouped into two categories, gradient-based methods and heuristic methods. Gradient methods are based on the computation of gradient vectors using the derivatives of the objective function to search for an optimum in the design space [13]. Heuristic methods are developed by taking inspiration from the observation of natural selection. These are also called stochastic optimization algorithms that can be used to analyze problems described by a set of random variables having known probability distributions. Heuristic search algorithms have achieved increasing popularity among researchers. An evolutionary algorithm or genetic algorithm is an example of a heuristic method that uses random choice as a tool to search for optimum through design space. A GA is a heuristic search algorithm used to solve optimization problems that relies on Darwin's theory of evolution. The application is based on evolutionary logic aimed at solving specific problems. The problem can be formulated in genetic terms, determining the representation scheme; determining the fitness measure; determining the parameters and variables for controlling the algorithm; determining the way of designating the result and the termination criterion [14].

In practice, the representation scheme is the way of mapping the problem variables, including the definition of a possible search space of the solution. Each variable of the problem is coded to become part (gene) of a unique chromosome, that is the genetic representation of a possible solution (individual), and the range of variability of each gene is defined in order to limit the exploration field of practical solutions. After the genetic formulation of the problem variables, the definition of a proper fitness measure is necessary. In the artificial world of mathematical algorithms, we measure fitness in some way and then use this measurement to control the application of the operations that change the structures in our artificial population. This value expresses the 'goodness' of generated solutions (individuals) based

on a well-defined performance parameter and the respective evaluation criterion. Thus, a set of algorithm parameters must be defined. In conventional genetic algorithms, they are the population size, the maximum number of generations to perform, and the percentage of use of genetic operators (selection, crossover, mutations, etc.). Finally, a termination criterion should be defined when the existence of the best solution, and therefore a best (or sub-optimal) reference performance values, were known. Otherwise, the algorithm can be manually stopped until a satisfactory solution is produced. Heuristic evolutionary solvers are used when optimization problems have many variables, and an optimal solution cannot be found through exact solvers. Every approach has drawbacks and limitations. In the case of Evolutionary Algorithms, these are extremely low, and the algorithm does not guarantee a solution.

Algorithms

The algorithms, and more generally computational techniques, can be applied with success to a wide range of structural and architectural problems. An algorithmic approach to handle complex problems is a powerful strategy to find optimal solutions. GA is one of the most efficient methods for the arrangement of the optimization problem of form-finding. GA is defined as a "search algorithm based on the mechanics of natural selection and natural genetics"; [14]. GA was developed by John Holland and his team at the University of Michigan in the 1960s. At present, GA provides a robust, efficient, and flexible tool to solve complex problems, such as air-traffic programming, weather forecasts, share portfolio balance, and electronic circuits design, in which a combined analytic way of resolution is unknown. Moreover, as far as the world of construction is concerned GAs are rarely used to deal with the optimization of large-span structures, the form-finding of shells, membranes, and the spatial configuration of reciprocal frames. GA has such a distinctive feature that it can easily handle discrete variables, in addition to the best in large and medium, it can also obtain multiple medium and good solutions at the same time [15]. These characteristics of the genetic algorithm are especially important when used as a structural design tool. This technique has been mainly chosen for its heuristic, or meta-heuristic, search method of the solution, which suggested, especially with reference to early conceptual design stages, to integrate the concept of structural engineering in the early architectural design process.

Relaxation algorithms were initially developed as tools for the approximate calculation of structures but, after a few years, other similar procedures, such as the Force Density Method were also studied with the aim of developing essential form-finding tools to support the architectural and

structural design of tensile structures. Nowadays, the design community has largely focused on the study of dynamic methods with special attention to Dynamic Relaxation, and Particle Spring Systems. Dynamic methods may be considered as all types of numerical techniques based on discrete sets of interconnected, or disconnected, point masses called particles, or sometimes nodes, permitting to describe the dynamic behavior of a system over a finite period regardless of the type of dynamic schemes and numerical integration adopted. DR is a numerical method for solving systems of ordinary differential equations using explicit time integration scheme damping techniques. DR is focused on finding a static equilibrium of a structure exhibiting material and geometric nonlinearities. The method directly uses Newton's Second Law of Motion and when damping is included in the formulation the motion eventually comes to rest giving a static analysis. DR was developed by Alistair Day in 1965. This method requires the discretization of a continuum into a finite set of nodes with lumped masses that are interconnected via elements and each element corresponds to a topological subset of connected nodes in which forces are calculated through its links. These calculations only depend on the position of nodes and are governed by Hooke's law of elasticity. The advantage of using D.R for form-finding during a design process is that changes can be made during the analysis and if used in an interactive form the method allows the designer, to alter the structure during the design to comply with the design constraints.

By using computational optimization methods, we aim to improve both architectural and structural design quality and reduce the risk of errors in the design process, which shortens the design time and reduces engineering cost and allows greater freedom when using conventional software. An application of design approach and structural optimization has been conceived by referring to the project AMA Roma (Municipal Urban Hygiene Company) Ecological Island: The Ecological islands are infrastructures open to citizens' equipment for the temporary storage of non-organic waste, organized in a different manner, to be subsequently undertaken for recycling activities.

Application and results

The roof structure of Ecological Island (Fig. 1) has been completely redesigned with the aid of a GA and a DR. The new design of the roof cover had to satisfy constraints due to functional needs, considering the differentiated height of supports, in order to accommodate vehicles for loading/unloading operations or reduce the number of supports, for facilitating vehicle travel. The original roof cover was too thick and rigid for this aim, we want to propose a new possible hypothesis of a solution that allows us to imagine a realizable form, which was studied by using an optimization procedure through algorithms in the software Rhinoceros/Grasshopper.

The design development of this structure was handled through a computational modeling workflow consisting of two stages. First, the generation of tentative shapes in the CAD software with defined algorithms. Next, the spatial configuration of each individual has been transferred to the FE program, which evaluates the stability of the structure. The conceptual design stage uses exploratory GA and DR for conducting exhaustive form-finding studies of the global geometry of the roof structure. The design evaluation stage then takes the results that best satisfy functional and formal/aesthetic criteria to produce FE-models and rapidly check structural capabilities. If the chosen design approach does not match structural requirements, the exploratory process is started again. Once a mature solution is reached, in which all the design criteria are satisfied, the analysis stages produce a detailed FE model to run a complete analysis of the structure. The initial dimensions of the roof cover were 33×33 m square steel structure with open space inside, the roof was built of a steel truss with 22 steel columns. The architectural concept of the new design of roof cover was inspired by the upcycling logo to stand for reuse not to recycle. The roof structure was modeled on the shape of a logo. The process starts by discretizing the NURBS geometry of the roof system (Fig. 2).



Fig. 1 Ecological island, waste collection center, Rome, Italy

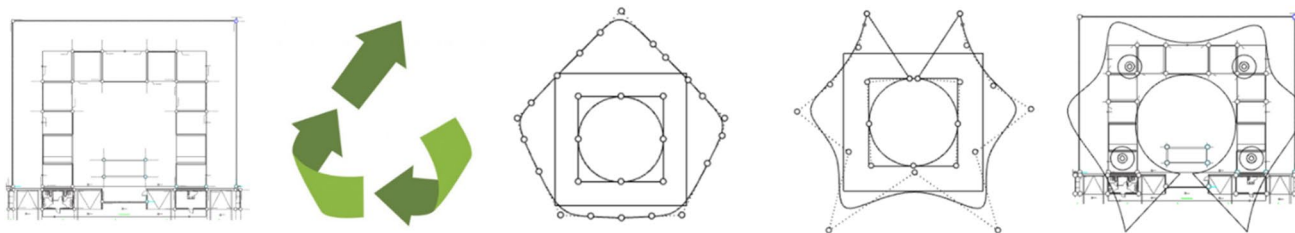


Fig. 2 The architectural concept of the roof cover

The shape of the shell is described by means of NURBS surface representation with a net of 4×21 control points in the U V direction as shown in Fig. 3. The vertical positions (z coordinates) of NURBS surface control points are assumed as the design variables. Some of these points, corresponding to the pillar-slab joints, have been then considered as fixed to preserve the heights of the roof supports. In this case, the roof maximum height, which corresponds to the design variable maximum extension, has been obtained in relation to its functionality, a minimum height of 5.6 m must be guaranteed in correspondence of the handling areas of the containers to accommodate vehicles for loading/unloading operations. So, the domain range of 2 m is given for roof cover as a search space for the algorithm to find the optimal shape. The total strain energy has been chosen as an objective/fitness function that must be minimized. The position and the height of pillars, as well as the plan projection of the roof boundary and its thickness of 15 cm, have been therefore identified as boundary conditions of the problem. The support system for the roof cover has been inspired by a tree column at four corners of the roof, where each column has four branches. So, the number of supports on the ground is reduced from 22 to 16 (4 columns x 4 branches).

This value has been chosen for functionality reasons, related to the architectural proportions of the building, considering that the final shape should never exceed the height of supporting pillars, or of the covered free space. To define

the objective function and arrange the optimization problem, total strain energy must be calculated. To this aim, we used a Rhino-Grasshopper plug-in called Karamba 3D, which is an interactive, parametric finite element program developed by Clemens Preisinger in cooperation with Bollinger und Grohmann ZTGmbH Vienna. It allows you to combine the definition of a parametric geometric model with finite element calculations and optimization algorithms like Galapagos to search for shapes. Nevertheless, since NURBS control points do not generally lie on the surface that they define, to introduce this restraint the control points net has been changed into an interpolating points net. From interpolating points, a surface has been created and modified into a mesh (50×50) as in Karamba3D shells are based on Grasshopper meshes. By using the “MeshToShell”-component which takes a triangle or quad mesh and turns it into a group of shell elements Quads get automatically decomposed into triangles, we obtain a shell structure. Shell patches are rigidly connected when some of their nodes have the same index [16]. To define the support system, four closest points are randomly chosen from the defined mesh vertices at each column that connects to the ground. A nodal load of 2 kN is applied on the vertices of the mesh in the negative z -direction and one gravity-load has been combined to form a load case. To calculate the behavior of a real-world structure, the Karamba analysis component computes the deflection for each load case and adds this information to the model. The analysis component also outputs the maximum nodal

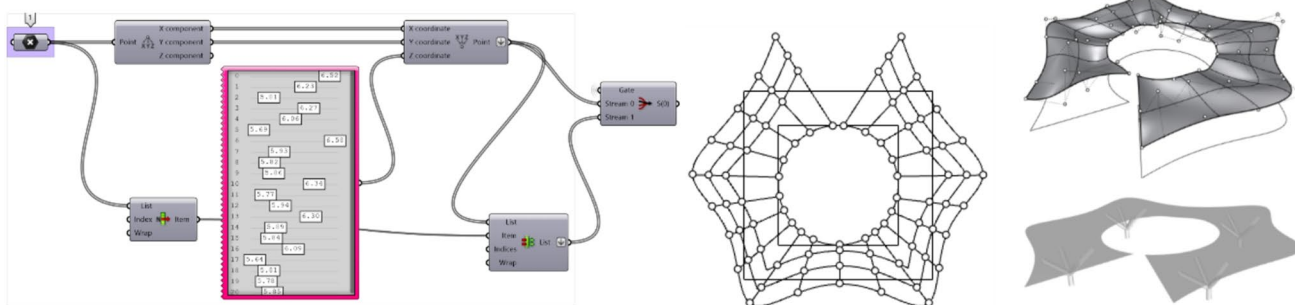


Fig. 3 Grasshopper definition and NURBS surface control points representation

displacement, the maximum total force of gravity, and the structure's internal deformation energy of each load case for details on work and energy. Thus, the values are used to rank the structure during a structural optimization procedure. The more efficient a roof structure the smaller the maximum vertical displacement, the amount of material used, and the value of internal elastic energy. Finally, to solve the optimization problem and run the genetic algorithm we used an optimization algorithm embedded in Grasshopper called Galapagos which is an Evolutionary solver. Galapagos has two inputs: Genome and Fitness. A fitness value can be described as the value you want to minimize, in this case, the total strain energy. The Genome is a collection of parameters that influence fitness, in this case, the vertical position (z coordinates) of the NURBS control points (Fig. 4).

The shape has been optimized with a GA, in which the total strain energy is the objective function to be minimized and z coordinates of the NURBS surface control points are the design variables. The evaluation of the result given by the software has not matched structural requirements. However, the design iteration has not been considered because the purpose of this application was not related to tool efficiency but, on the contrary, on the experimentation of structural efficiency itself. The main issue that arose was the shape, which is especially important in both architectural and engineering problems. It was complicated to design because the shape was quite inflated around the edges of the roof system. There is to be a translation from the NURBS curve of the external boundary and to the circular curve of the internal space,

the interpolation between the NURBS curve and the circle is quite problematic. Another issue was structural complexity in the shape which will be difficult to construct on-site. To solve this problem, two methods can be considered: first to change the continuous shell into a grid shell, and second to change the design approach. To this aim, first, we decided to change the continuous shell into a grid shell which combines the form with the efficiency of a structure driven by construction simplicity. For this reason, a Kagome lattice (Aniso grid) consisting of triangles and hexagons has been chosen to project on the free form surface generated by the Evolutionary Solver. The lattice structural grid was developed using the Lunch Box plugin in grasshopper. The process is taking a free-form surface, paneling it with hexagonal cells. It generates a flat list of hex cells and by joining the midpoints of the sides of the hexagon cell, we obtain a Kagome lattice grid as shown in Fig. 5.

Aniso grid structures can be consistent with different patterns and are not limited to the equilateral triangle. In other words, they can be non-isotropic in terms of structural patterns. To analyze the structure and check the structural stability, the model has been exported to the FE program which is Dlubal RFEM with help of the Rhino/Rfem common interface. The plugin RhinoRfem connects the 3D graphical software Rhinoceros/Grasshopper and the structural analysis program RFEM by Dlubal. In order to reduce the computation time, we have analyzed half of the structure as it is symmetric. However, the performance of the structure was not good enough to withstand the loads shed on the roof

Fig. 4 NURBS representation and GA outcome of the roof cover

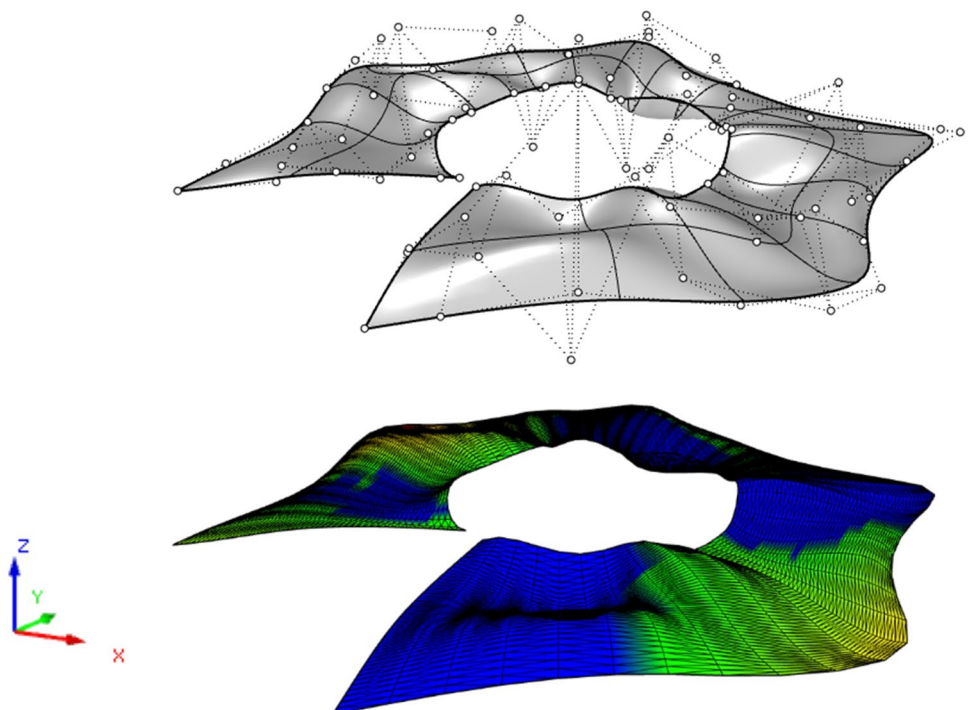
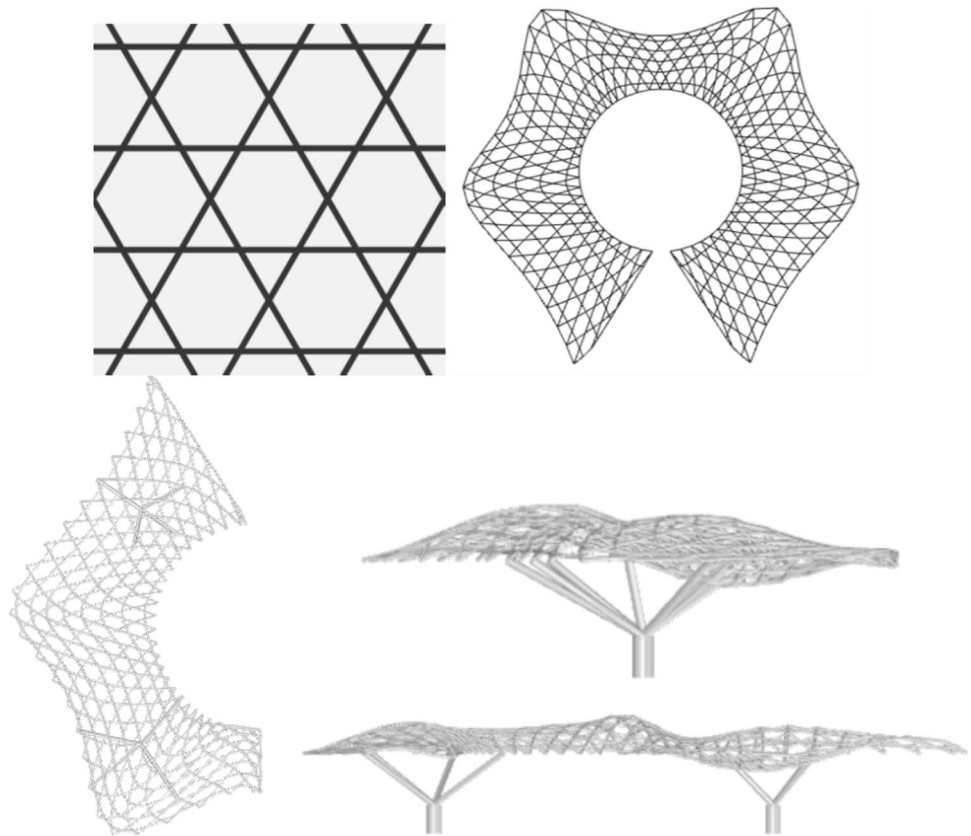


Fig. 5 Kagome lattice grid pattern and resulting roof cover



system. As mentioned earlier, If the chosen design approach does not match structural requirements, the exploratory process is started again. So, we decided to change the design approach. This time we chose a dynamic relaxation approach to finding the optimal form.

To this aim, the form-finding process was integrated into this workflow through the introduction of the Kangaroo. Kangaroo Physics, which was developed by Daniel Piker, Kangaroo is to date the most popular tool within the large community of designers using Rhinoceros for integrating physical behaviors through fast simulations within the modeling process. Piker described Kangaroo as a physics engine directly embedded in the parametric modeling environment of Rhinoceros-Grasshopper allowing interactive exploration of geometrical shapes through simulated behaviors based on material properties and applied forces. Kangaroo is a physics engine for Grasshopper. A physics engine is a collection of algorithms that enable a computer to simulate some aspects of the behavior of real-world objects [17]. The shape of the roof was derived by defining a single surface and then relaxing a grid over the surface. The design process starts by describing the boundary NURBS curves of the roof cover. The boundary curves are manually described in the CAD software. To arrange the physics engine, we need to define the goal objects which are Anchor, Length and Load. The anchor will keep a point in

its original location and length tries to keep two points (line endpoints) at a given distance from each other and load is a force specified as a vector and the length of the vector is the magnitude of the applied load. Once the goal objects are defined Kangaroo solver is ready to solve the system. The Form Finding process has begun with an assumption of four support systems. The surface was form-found through a network of springs in Kangaroo which balanced the equalization of spring lengths (surface relaxation) with an upwards load vector like the inverse hanging-chain model. It outputs the solved vertex locations, by taking those vertex points a surface has been created to project a lattice grid as shown in Fig. 6. Then a flat Kagome pattern is projected on the free form surface that is obtained from the relaxation method, and then the structure is transferred to the FE program in order to verify the structural performance.

The form-finding described above, finding the optimal shape for a shell for desired or needed constraints and goals is essentially an optimization process, usually followed in a trial-and-error manner. However, the evaluation of this iteration with 4 support systems does not match the structural requirements, to solve this problem we progressively increased the number of supports. So, after many iterations and changes in the support system and shape, the final desired efficiency was achieved with six support systems (Fig. 7).

Fig. 6 Formation of grid shell through surface relaxation
a Mesh creation from anchorage points, **b** Kagome grid pattern projection on mesh, **c** Final grid shell

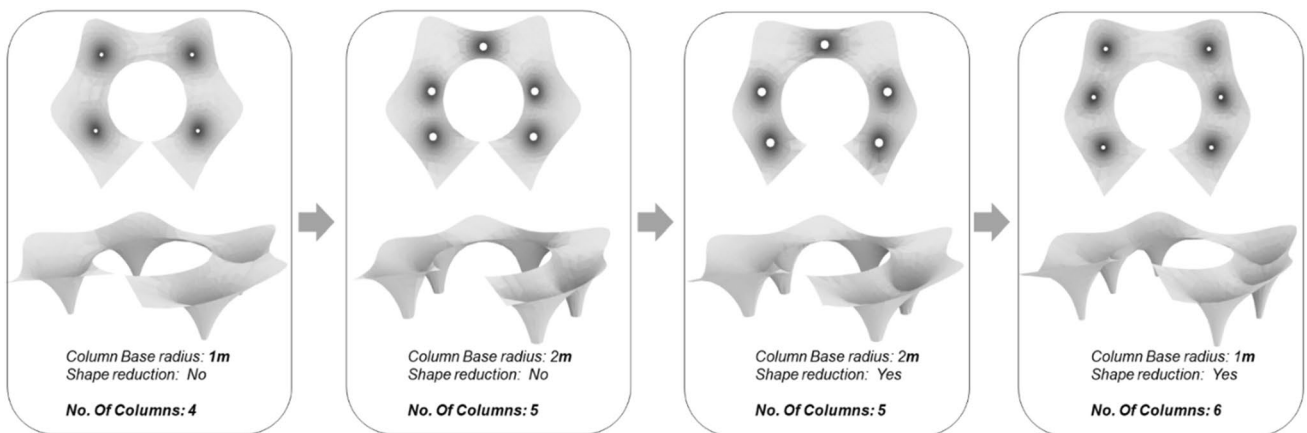
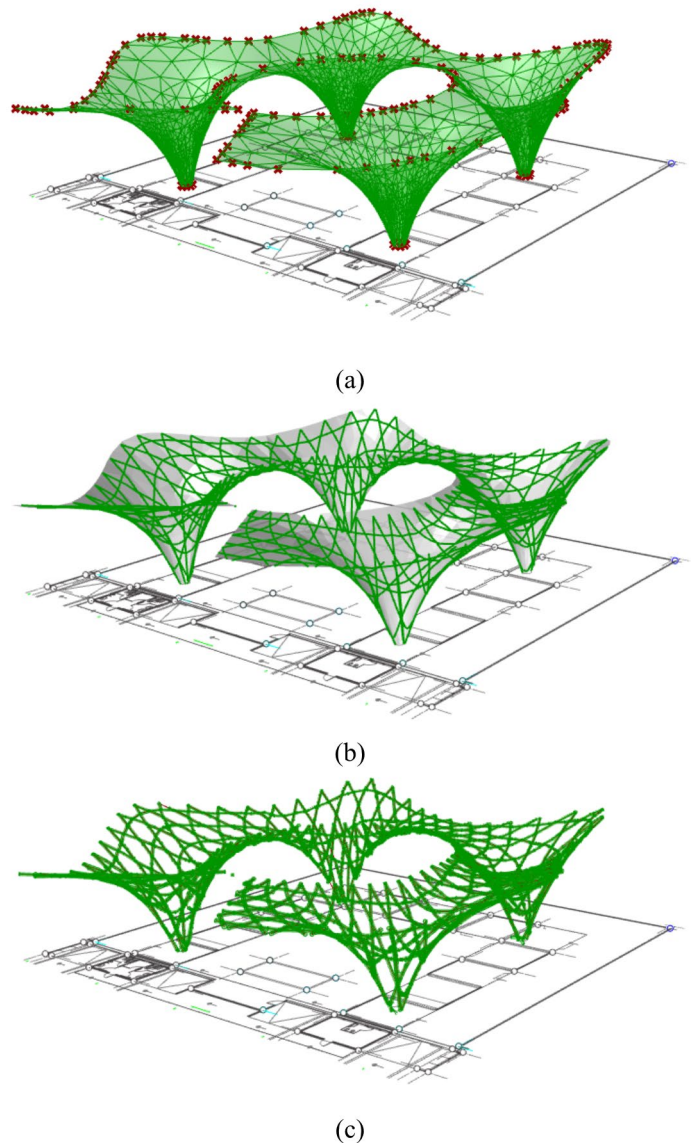


Fig. 7 Evolution of form-finding process of the roof cover

After conducting many iterations of exploratory design approaches and modification of models through numerical simulations, the global geometry of the structure has been evolved and form found with the DR model in the conceptual design stage by considering the results obtained from the design evaluation stage which satisfied both functional and structural capability. Now, in order to run the complete FE Analysis, we use the structural analysis program Dlubal RFEM. To analyze and evaluate the performance, the model has been exported from CAD to FE program with 1768 Beam elements. A Steel S355 circular pipe 100 mm diameter and 10 mm thick cross-section have been chosen for beams and a 2 mm thick tensile membrane as a cover for the grid. A load is applied in a negative z -direction to all intersecting nodes of the grid structure. In addition to the self-weight, the calculation included wind load cases, the load situations were provided by the Dlubal software online

GEO-ZONE Load Determination tool and wind velocity values are inserted manually in the software. The assessment of serviceability and performance under dynamic loads will be performed in a forthcoming work.

After running the simulation in RFEM, the results of the roof cover are provided with global deformation of 167.4 mm for beams and 57.2 mm for the surface (Figs. 8 and 9). The optimal results obtained from tests show that the optimization process based on the Relaxation Algorithm converges more quickly than the GA, but also gives some information on the way both algorithms explore the feasible solution domain (Fig. 10). The optimization path naturally depends to a great extent on the choice of the algorithm parameters, and in particular on the way the search domain is adapted through the optimization process in order to improve the convergence speed. Therefore, in this application, the development of GA efficiency has not been considered.

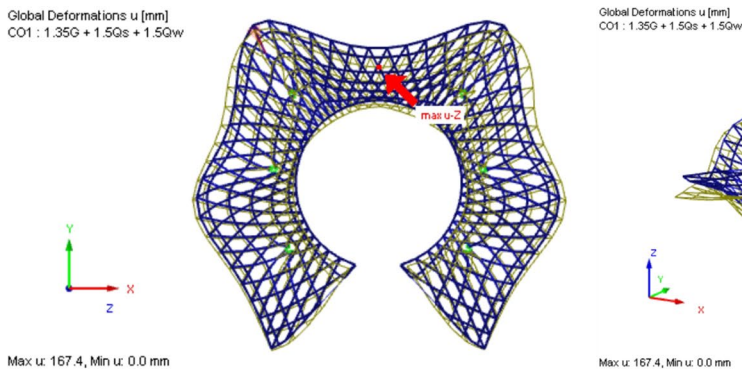


Fig. 8 FE results of the grid shell obtained in RFEM

Fig. 9 FE results of the roof surface obtained in RFEM

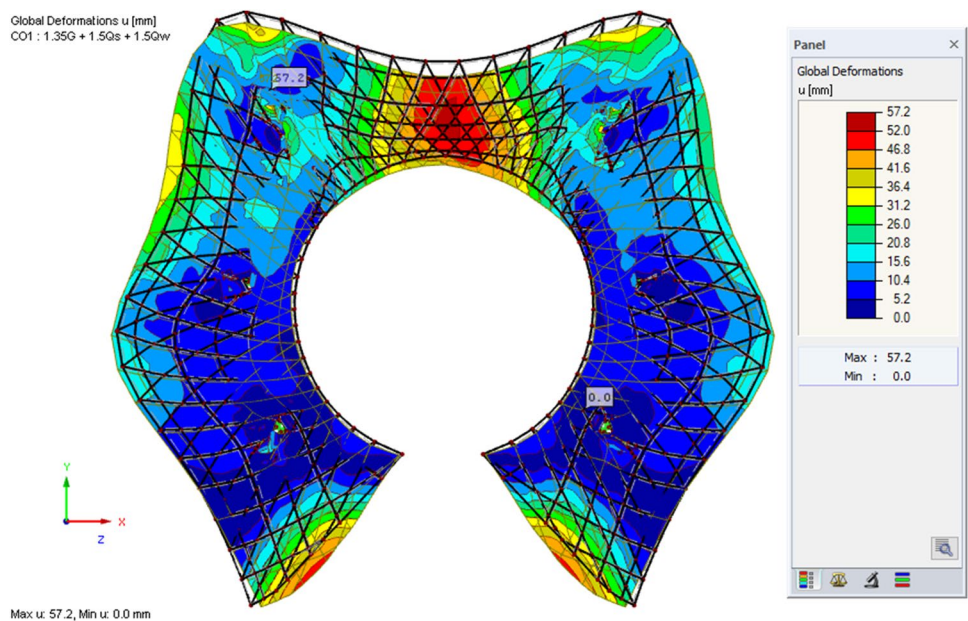
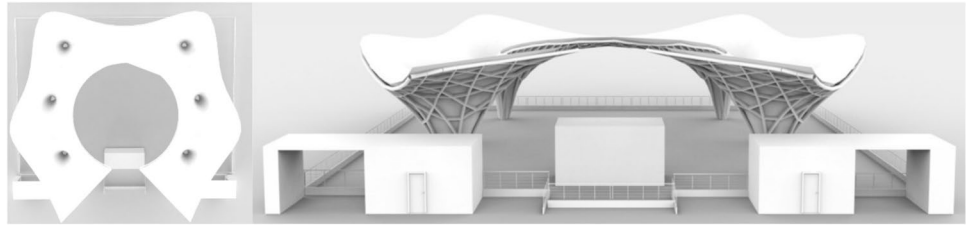


Fig. 10 Final design outcome of the roof cover



The choice between different computational methods becomes crucial when dealing with shape resistant structures. Where it is not possible to separate the ‘form’ and ‘force’ in order to maintain formal characteristics and structural efficiency. In contemporary architecture practice, there are interesting examples in which designers aim to develop computational methods for form-finding optimization to realize complex geometry structures. For instance, The Crematorium of Kakamigahara, Gifu, Japan 2006 designed by architect Toyo Ito and structural engineer Mutsuro Sasaki [3]. The structure is simply composed of a free-form concrete roof, supported by a set of columns randomly positioned on the ground level, this roof form, which perfectly integrates architecture and structure, is first defined by the architect, and then optimized through shape sensitivity analysis leading to satisfying the form and force.

Final remarks

The principal aspects of this work focused on the close relationships among topologic, geometric, and numerical conditions built based on algorithms for enabling more intuitive design practices promoting modern design approaches for user-model interaction. This should be considered in the context of an exploratory new computational method for form-finding that raises innovative design problems for shape exploration and computation. This condition does not only imply the development of more intuitive modelling workflows but most importantly a deeper understanding of rules and conditions that could allow reliable topological transformations on numerical models. Computational tools used for design approaches introduced in this work could significantly improve conceptual design exercises in practice, as a way to generate and compare a wide range of design ideas quickly and easily and evaluate the structural efficiency. In a more integrated approach, a team of architects and engineers could collaboratively develop alternative designs that perform well structurally and achieve architectural design goals. Through the integration of numerical simulations within earlier design stages, such performative criteria can be quickly evaluated to assist the decision-making for structural design. On this subject, the growing interest in further

exploring the complex interrelation between their form and structure is gradually requiring more computational avenues to interact with physical behaviors, especially during conceptual design stages.

In summary, the described workflow is best suited for speedy exploration of design approaches with a qualitative understanding of structural behavior, maintaining an awareness of downstream implications of design operations, enabling a reasonable correspondence between aesthetic quality and structural efficiency, and CAD geometries. The Rhino RFEM interface was a better choice for this initial research, connecting CAD geometry and the FE program which made this design workflow seamless. The design approaches are flexible and were successfully applied to the roof structure, with quite different constraints. To formulate proper constraints, an important aspect is careful consideration of how the structure will be constructed and fabricated. It is also important to consider that the resulting discrete wireframe model still needs to be dimensioned and materialized, evaluating the structural performance using FE analysis could significantly improve the conceptual design. Although this preliminary investigation has given some useful insights into the possibilities of improving the conceptual structural system, further studies are needed to consider different architectural and structural requirements.

Acknowledgements The Authors gratefully acknowledge the financial support provided by the Italian Ministry of University and Research P.R.I.N. National Grant No. 2017HFPKZY (B88D19001130001), Sapienza grant (Progetti Grandi 2021, B85F21008380001).

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Declarations

Competing interest The Authors declare that they have no financial interests related to the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not

permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Mueller CT, Ochsendorf J (2013) An integrated computational approach for creative conceptual structural design. In Proceedings of the IASS Annual Symposia
- Adriaenssens S, Block P, Veenendaal D, Williams C (eds) (2014) Shell structures for architecture: form finding and optimization, 1st edn. Routledge. <https://doi.org/10.4324/9781315849270>
- Sasaki M (2005) Flux structure. Toto Publishers, Tokyo
- Benvenuto E (1991) An introduction to the history of structural mechanics, 3rd edn. Springer-Verlag, New York
- Trovalusci P, Panei R (2010) Towards an ethic of construction: The structural conception and the influence of mathematical language in architectural design. ICSA2010- Proceedings of 1st International Conference on Structures and Architecture. Taylor & Francis Group, London, pp 180–187
- Trovalusci P, Tinelli A (2013) The recovery of the ethic of constructions: P. L. Nervi vs. S. Musmeci, two structural conceptions compared. ICSA2013 - Proceedings of the 2nd International Conference on Structures and Architecture. Taylor & Francis Group, London, pp 261–267
- Trovalusci P, Tinelli A (2010) Structural optimization vs. shape design. ICSA 2010- Proceedings of the 1st International Conference on Structures and Architecture. Taylor & Francis Group, London, pp 116–123
- Panei R, Trovalusci P, Tinelli A (2016) The question of the technique: From the designing idea to the realized form, ICSA2016 - Proceedings of the 3rd International Conference on Structures and Architecture. Taylor & Francis Group, London, pp 519–526
- Xie YM, Steven GP (1997) Evolutionary structural optimization. Springer-Verlag, Berlin, pp 40–61. <https://doi.org/10.1007/978-1-4471-0985-3>
- McNeel R, others (2010) Rhinoceros 3D, version 6.0. Robert McNeel & Associates, Seattle, WA
- Rutten D (2010) Evolutionary principles applied to problem solving. Grasshopper algorithmic modeling for rhino. www.grasshopper3d.com/profiles/blogs/evolutionary-principles. Accessed 26 June 2021
- Tedeschi A, Andreani S (2014) AAD, Algorithms-aided design: Parametric strategies using grasshopper. Le Penseur Publisher
- Kirsch U (1981) Optimum structural design: concepts, methods, and applications. McGraw-Hill, New York
- Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., USA
- Koza JR (1994) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge
- Preisinger C (2013) Linking structure and parametric geometry. Archit Des 83:110–113. <https://doi.org/10.1002/ad.1564>
- Piker D (2013) Kangaroo: form finding with computational physics. Archit Des 83:136–137. <https://doi.org/10.1002/ad.1569>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.