

Atypical design optimisation of structures: case studies

A. Maheri

Faculty of Engineering and Environment, Northumbria University, UK

Abstract

This paper presents three case studies on structural optimisation. In the first case study a nonconventional approach is adopted to deliver an integrated design of trusses. This integrated design method, compared to traditional integrated design methods, while is significantly less demanding in terms of computational power, produces reasonable solutions, superior to sequential designs. This is realised by defining a new parameter connecting design qualities in one domain to design variables in another. The second case study, defined on topology (aerodynamic) optimisation of wind turbine blades, describes a nonconventional crossover method, allowing incorporating more domain specific knowledge in the search. The third case study explains a software tool developed for optimisation of cellular structures, taking the advantage of high fidelity finite element analysis in ANSYS, integrated in a genetic algorithm.

Keywords: integrated domain design, load conversion expediency, geometric crossover, soft coding, structural optimisation.

1 Introduction

Any design optimisation problem, irrespective of its type, has three attributes, namely (i) design methodology (e.g. integrated, coupled, multidisciplinary, deterministic, etc), (ii) optimisation technique (gradient based, exhaustive search, heuristic search, meta-heuristic search, etc), and (iii) the software tool used for search and evaluation (in-house, commercial, low fidelity, high fidelity, etc). These features are highly interconnected, changing one normally affects the feasibility, performance or validity of the others (Figure 1). This dependency, while hindering designers to adopt a generalised approach in design optimisation, has been always a motivation for researchers to enhance existing



methods, techniques and tools. Among the vast amount of publications dealing with enhancing existing design methods (e.g. [1–4]), optimisation techniques (e.g. [5–7]) and software tools (e.g. [8–11]), the majority belongs to optimisation techniques. This paper presents three case studies on structural optimisation solved by nonconventional methods. The case studies are defined on topology optimisation of trusses supporting distributed loads, wind turbine blades and cellular structures.

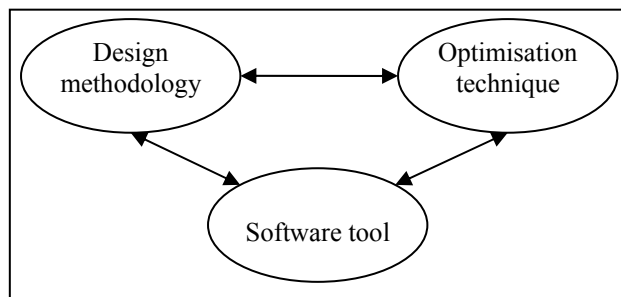


Figure 1: Three attributes of a design optimisation problem.

2 Innovative design methodology: integrated domain design using load conversion expediency

Topology optimisation of trusses can be divided into two categories with respect to the type of the external load: (i) problems in which the external load is originally concentrated with a known location given in the design specification and (ii) problems in which the external load is distributed in nature and is transferred to the truss either directly or via an interface. The interface can be another continuous body, such as a beam with negligible weight compared to the load, or can be a series of axially loaded members, such as hydraulic jacks, struts, cables, etc. In case of trusses supporting distributed loads, the load-bearing node distribution becomes a design variable affecting both the distributed load domain and the truss domain. That is, an optimum solution in truss domain might not be a global optimum solution when both truss and load domains are considered together. To achieve a global optimum solution, an integrated design optimisation must be carried out. On the other hand, conducting an integrated design might not be always feasible due to (i) significant increase in computational time corresponding to the evaluation of design qualities in the load domain (e.g. a full FEA of a complex body exerting the load) and/or (ii) lack of enough data for modelling and analysis of the load domain (e.g. in case of designing a multi-purpose support truss). In view of these shortcomings, Maheri and Maheri [3] proposed a robust method for topology and geometry optimisation of trusses supporting distributed loads, giving due considerations to the load domain. Their method for integrated design of trusses supporting distributed loads is based on introducing a new parameter, Load

Conversion Expediency (LCE) that can be treated as an additional design objective or constraint in truss design problems. LCE is defined as:

$$LCE = 1 - 2 \int_0^1 g(x^*) |I(x^*) - S(x^*)| dx^* \quad (1)$$

in which I and S are, respectively, the normalised cumulative distributed load and the cumulative concentrated forces and are given by the following equations:

$$I(x^*) = \int_0^{x^*} q^*(\eta) d\eta \quad (2)$$

$$S(x^*) = \left(\sum_{j=1}^i F_j^* \right) \left(< x^* - a_i^* >^0 - < x^* - a_{i+1}^* >^0 \right); \quad (3)$$

for $a_i^* \leq x^* < a_{i+1}^*$; $i = 1, 2, \dots, N_L - 1$

Unit step function $< x^* - a_i^* >^0$ is 0 when $x^* < a_i^*$ and 1 when $x^* \geq a_i^*$. The normalised quantities are defined as $q^* = q/q_{\max}$; $x^* = x/L$; $a^* = a/L$ and $F^* = F / \int_0^L q(x) dx$.

In the above equations, L is the length of the truss domain, N_L is the number of load-bearing nodes, q represents the distributed load and a_i stands for the x -coordinate of the node on which the nodal concentrated force F_i is applied.

Function g reflects the contribution of those parameters exclusive to the load domain and that affect the stresses and deformations in that domain:

$$g(x^*) = \text{func}\{\mathcal{H}^*(x^*), \mathcal{M}^*(x^*)\} \quad (4)$$

Parameters $\mathcal{H}^*(x^*)$ and $\mathcal{M}^*(x^*)$ represent the set of the relevant normalised cross sectional properties and the set of the relevant normalised material properties of the load domain, respectively.

Since $g(x^*)$ depends on the normalised cross sectional and material properties, it can be found without having the exact values of $\mathcal{H}(x)$ and $\mathcal{M}(x)$. For example $g(x^*) = 1$ represents the case of a load domain with homogenous material $\partial \mathcal{M}(x) / \partial x = 0$ and uniform geometry $\partial \mathcal{A}(x) / \partial x = 0$.

As Equation (1) suggests, LCE depends on the form of the distributed load, the number and location of the load-bearing nodes and the normalised values of the contributing parameters from the load domain not the actual values. Maheri and Maheri [3] showed that load-bearing node distributions which maximise LCE are all in close proximity of the load-bearing node distributions which minimise maximum stress, maximum deflection and maximum reaction force in the load domain and then concluded that LCE can be used as a reliable substitute for other objectives in a semi-deterministic integrated design. The following case study shows how one can perform an integrated design with minimum knowledge of one of the domains.

2.1 Integrated domain design using load conversion expediency case study: design of a platform for wind turbine blade enhancement

It is desired to design a multi-purpose platform for inspection, enhancement (installing aerodynamic control surfaces, anti-icing blankets, etc) and system repair of 20-meter long wind turbine blades. Due to the long span of the blades and the multiple function of the platform, the design concept of Figure 2 was selected for further development. It is essential to control the deflection of the blade as well as the contact forces applied on the blade by hydraulic jacks, while keeping the number of hydraulic jacks to a minimum value. In the truss domain, the only constraints are the location of the supports $(0,0)$ and $(0,L)$, number of load-bearing nodes $N_L = 5$, and the x -coordinate of the left and right load-bearing nodes $x=0$ and $x=L$ respectively. As a result of using hydraulic jacks, excessive nodal displacement will not be an issue here.

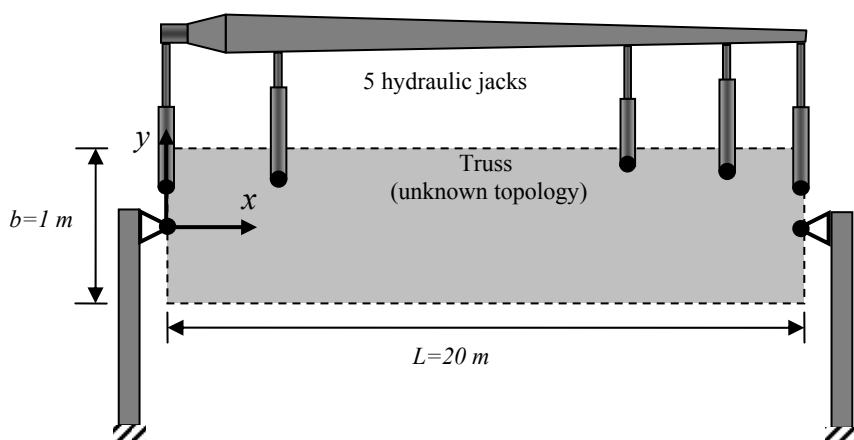


Figure 2: Design concept for a multi-purpose wind turbine blade platform.

The aim of this case study is to demonstrate the proposed integrated design method. Without loss of generality, the optimisation is focused on the truss topology and geometry. Sizing is not directly included in the optimisation at this stage. However, instead of assuming identical cross-sections for all members, it is assumed that the cross section of each member is proportional to its internal force. This is a valid assumption for members in tension and short members in compression and an approximation for long members in compression, leading, at this stage of design, to an approximated continuous size optimisation. Having the topology and geometry optimised, a discrete approach can be used afterwards for size optimisation, taking into account stresses in members, stability of members and nodal displacements.

Assuming that members are made of the same material and that their cross sectional areas are proportional to their internal forces, minimisation of the

weight of the truss is equivalent to minimising the objective function h , defined as

$$h = \sum_{i=1}^m l_i (\bar{X}) P_i (\bar{X}) ; i \in \{1, 2, \dots, m\}, \quad (5)$$

where, l_i and P_i ($P_i \neq 0$) represent the length and the internal force of member i respectively and m is the number of members in the truss. In the case of zero members ($P_i = 0$), P_i will be replaced with the smallest internal force in the truss members. In the above equation $\bar{X} = \{N, x_j, y_j, C\}$, ($j \in \{1, 2, \dots, N\}$), stands for the vector of design variables. Parameter N is the total number of nodes, C represents the connectivity matrix, and x_j and y_j are the coordinates of node j respectively. In this study, $N_L = 5$ and LCE is treated as an inequality constraint in truss design:

$$LCE \geq LCE_c \quad (6)$$

in which LCE_c is the smallest permissible (goal) LCE.

In order to solve the simple optimisation problem of Equation 5, LCE for each design candidate must be calculated. In order to calculate LCE, in addition to the location of the load-bearing nodes, normalised distributions of the weight of the blade, q^* , blade equivalent flexural rigidity, EI^* , and the equivalent concentrated load system F_i^* ($1 \leq i \leq N_L$) are required. Since normalised values are required, the LCE can be easily found without having the actual values of the distributed load and equivalent flexural as follows (see [3] for details).

$$q^*(x^*) = \frac{\rho g \oint_{\text{contour}} t_{\text{shell}} ds}{q_{\text{max}}} = \frac{(c^* t_{\text{max}}^*)^{1/2}}{\max \left\{ (c^* t_{\text{max}}^*)^{1/2} \right\}} \quad (7)$$

$$EI^*(x^*) = \frac{E \oint_{\text{contour}} y^2 t_{\text{shell}} ds - EI_{\text{min}}}{EI_{\text{max}}} = \frac{(c^* t_{\text{max}}^*)^{3/2} - \min \left\{ (c^* t_{\text{max}}^*)^{3/2} \right\}}{\max \left\{ (c^* t_{\text{max}}^*)^{3/2} \right\}} \quad (8)$$

Parameter c is the chord length, ρ stands for the material density, and t_{max} and t_{shell} are the aerofoil and shell thicknesses at span location x respectively.

To calculate the LCE, the equivalent concentrated load system F_i^* ($1 \leq i \leq N_L$) is also required. For each design candidate under evaluation, having the location of load-bearing nodes, the equivalent concentrated load system F_i^* ($1 \leq i \leq N_L$) can be easily found.

Using a genetic algorithm (see reference [3] for details), the optimisation problem of Equations (5) and (6) was solved for $LCE_c = 0.94$. Besides the semi-deterministic integrated design formulated above, the truss was also designed utilising a sequential design approach with two scenarios for load-bearing node distributions: (i) equally spaced load-bearing nodes and (ii) load-bearing node distribution based on the load distribution.

Results are shown in Table 1. As it is evident in this table, the solution obtained by semi-deterministic integrated design is superior to the solutions obtained by sequential design in all aspects. Comparing the approximated weights and the LCE of the solutions shows that the result of integrated design is superior to the results of sequential designs with respect to both the truss and the load domains. Treating the locations of the load-bearing nodes as design variables makes the truss design problem more flexible leading to better solutions. By selecting a realistic high value for LCE_c , any feasible solution which satisfies the constraint applied on LCE produces minor complications in the load domain. Since LCE is calculated via a simple numerical integration which requires negligible computational time compared to the truss solver module, a semi-deterministic integrated design virtually requires no more computational time than that required by simplistic sequential designs. It should be noted that the only information required from the load domain (wind turbine blade) was the normalised chord and the aerofoil thickness distributions.

Table 1: Optimisation results for the multi-purpose wind turbine blade platform.

	Sequential design		Semi-deterministic integrated design
	Equally spaced load-bearing nodes	Load-bearing node distribution based on load distribution	Free load-bearing nodes subject to constraint on LCE
Approx. weight $\sum_{i=1}^{N_m} l_i P_i$	4.166	4.050	3.288
Total length $\sum_{i=1}^{N_m} l_i / L$	3.132	3.138	3.128
Max. force in members $P_{\max} / \int_0^l q^* dx^*$	2.701	2.519	1.839
Min. force in members $P_{\min} / \int_0^l q^* dx^*$	0.274	0.111	0.033
LCE	0.936	0.933	0.940
Load-bearing node distribution (x_i / L)	{0, 0.25, 0.50, 0.75, 1}	{0, 0.224, 0.410, 0.613, 1}	{0, 0.086, 0.241, 0.761, 1}

3 More engineering optimisation: phenotype building blocks, geometric crossover and partial fitness

It is well recognised that the efficiency of a genetic algorithm in exploration and exploitation of the solution space can be improved by incorporating domain-



specific knowledge into the algorithm. In many real-world applications, the physics of the problem suggests heuristics that can be incorporated into the search and selection procedures. Whenever genetic algorithms are applied to such problems, domain specific knowledge should be considered in the design of the reproduction operators as well as the representation and selection. Domain specific knowledge can be broadly incorporated in selection and reproduction operators, when employing the concept of phenotype building blocks (PBBs). PBBs contain information at phenotype level. PBBs are either the components of a multi-component system, or different parts of a continuous system with different design qualities and/or evaluation measures. Identifying PBBs, partial fitness can be defined at phenotype level; hence more detailed information can be processed during the search, leading to enhancement of the search efficiency. However, in order to preserve, propagate and recombine good building blocks efficiently, building blocks should have a low probability of being disrupted by crossover. The crossover operation should be designed at phenotype level. Geometric crossover (GCO) is applied on the phenotype rather than the genotype.

3.1 Phenotype building blocks, geometric crossover and partial fitness case study: design of wind turbine blades

Aerodynamic design of wind turbine blades includes optimisation of the topology of the blade. Parameters such as rotor radius and the span-wise distributions of the chord length, pretwist and aerofoils define the topology of the blade and are treated as design variables. The design variables can be either distributed (chord, pretwist, shell thickness and aerofoil code) or single value (blade span). The design variables can also be categorised as either continuous (blade span, chord, pretwist and shell thickness) or discrete (aerofoil code). Using n design (precision) points for the distributed design variables (chord length, pretwist, shell thickness and the aerofoil code), the total number of design variables will be $4n+1$. Employing a mixed real-number/indexed encoding, each individual can be defined by a chromosome of length $4n+1$. The genotype chromosome is a string of real numbers as well as indexes.

Blade is a continuous structure, but its inner and outer sections have different functions and can be defined as two phenotype building blocks, as shown in Figure 3. The outer section of the blade is aerodynamically more efficient. That is, the power is mostly produced by the outer section of the blade. Whilst the inner section experiences greatest internal forces and the design is driven based on the structural demand. Since the inner and outer sections have different functions they can be assessed separately. The inner section can be assessed based on, for example, the maximum flap stress or the weight of the blade, while the outer section can be assessed based on the produced average power.

In order to investigate the performance of employing GCO, PBB and partial fitness in optimisation of wind turbine blades, blade of AWT27, a 2-bladed stall regulated wind turbine, is selected to be optimised for the pretwist and chord distributions. Starting with the same initial population, 2000 crossover operations

of each type (GCO and arithmetic crossover) is carried out. In performing arithmetic crossover, a roulette wheel constructed based on the fitness $fitness = P_{av}$ is employed, while in the case of GCO, two roulette wheels are constructed based on the partial fitnesses $PF_1 = P_{av}$ for the outer section and $PF_2 = 1/weight$ for the inner section. These roulette wheels are employed to select the parents.

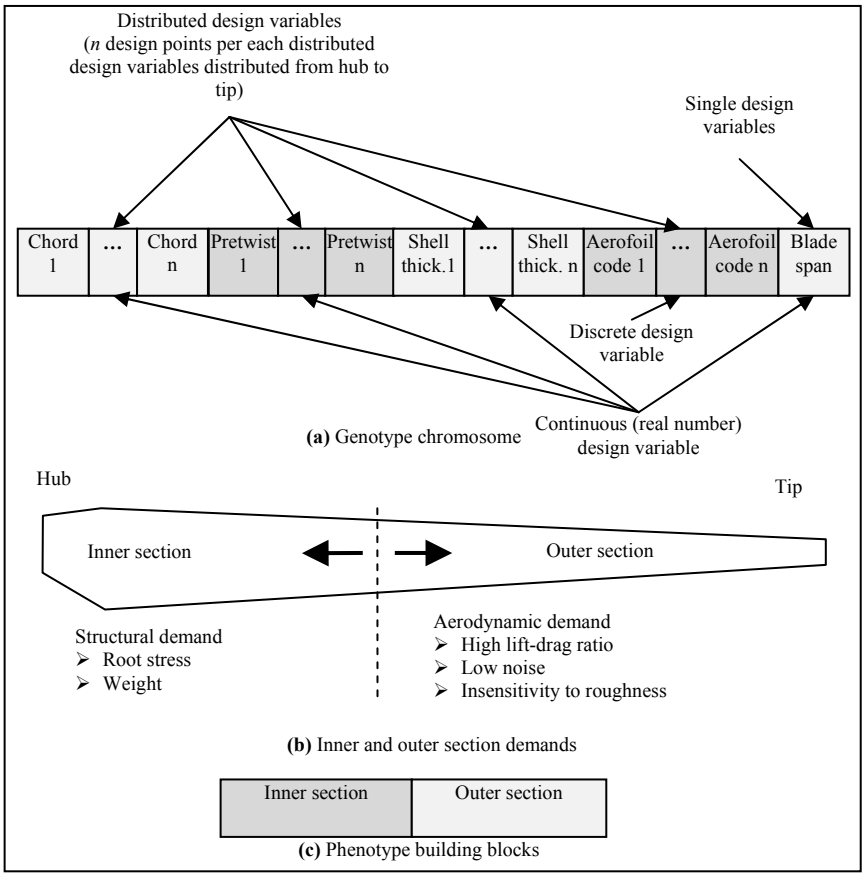


Figure 3: Wind turbine blade and its PBBs.

Results presented in Figure 4 show an improvement of 11.6% in the average power of the population when GCO is used versus an improvement of 8.6% when arithmetic crossover is used. Results also show that the population quality in terms of the weight improves slightly more when using GCO (12.2% versus 11.7%).

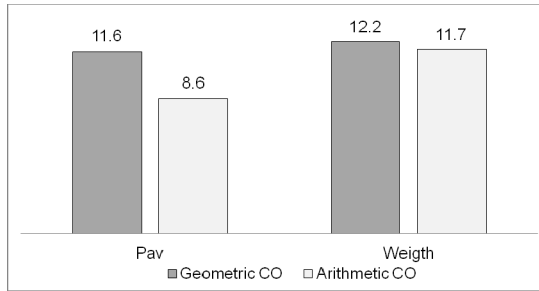


Figure 4: Percent improvement in the qualities of the population: geometric versus arithmetic crossover.

4 Robust tool: soft coded optimisers for high fidelity design candidate evaluation

In many design optimisation problems high fidelity evaluation of design candidate is essential to conduct a successful optimisation. Most of commercial packages suitable for high fidelity analysis of structures, such as ANSYS, incorporate some elements of optimisation. For instance APDL commands for killing or retrieving elements as required for conducting ESO (Evolutionary Structural Optimisation). However, the main and common drawback of these packages becomes highlighted when the design problem needs to be redefined (e.g. change in geometry, boundary condition and loading) within the optimisation process. In cases like this, irrespective of the robustness of the optimiser, high fidelity evaluation of design candidate can be a bottle neck affecting the overall performance of the optimisation process.

As a practical means to circumvent this problem, without jeopardising the quality of the search, one can adopt a soft-coding approach in modelling and problem definition as required for high fidelity analysis packages. Taking ANSYS as an example of a package for high fidelity analysis and MATLAB as a programming environment highly convenient for developing problem specific optimisation engines, a soft-coding approach refers to writing codes in MATLAB that generate APDL files which are required to model the problem in ANSYS.

Figure 5 shows two main modules of an optimiser adopting a soft-coding approach, namely the search engine and the evaluator. The search engine, written in MATLAB, generates a set of design variables $\{\vec{X}\}$, which together with the set of (fixed) design parameters $\{\vec{X}_0\}$ define a design candidate. The union of these two sets includes all parameter (geometry, size, topology, patches, material, load and boundary conditions) required for definition and modelling of the problem in ANSYS. After creating a design candidate, ANSYS also needs solution setup parameters. While some solution setup parameters $\{\vec{S}_0\}$ are fixed and can be defined outside of the search engine, some others $\{\vec{S}\}$ can be problem dependent

and therefore should be defined based on the features of the design candidate at hand (e.g. mesh size, element type, large or small deformation, etc). Having all data required for modelling and solving the problem, the APDL files are generated by MATLAB as the first part of the evaluation module and then ANSYS is called (through MATLAB) to read the APDL files, analyse the design candidate and write the analysis results. In the next stage the analysis results are read in MATLAB and the design candidate is assessed.

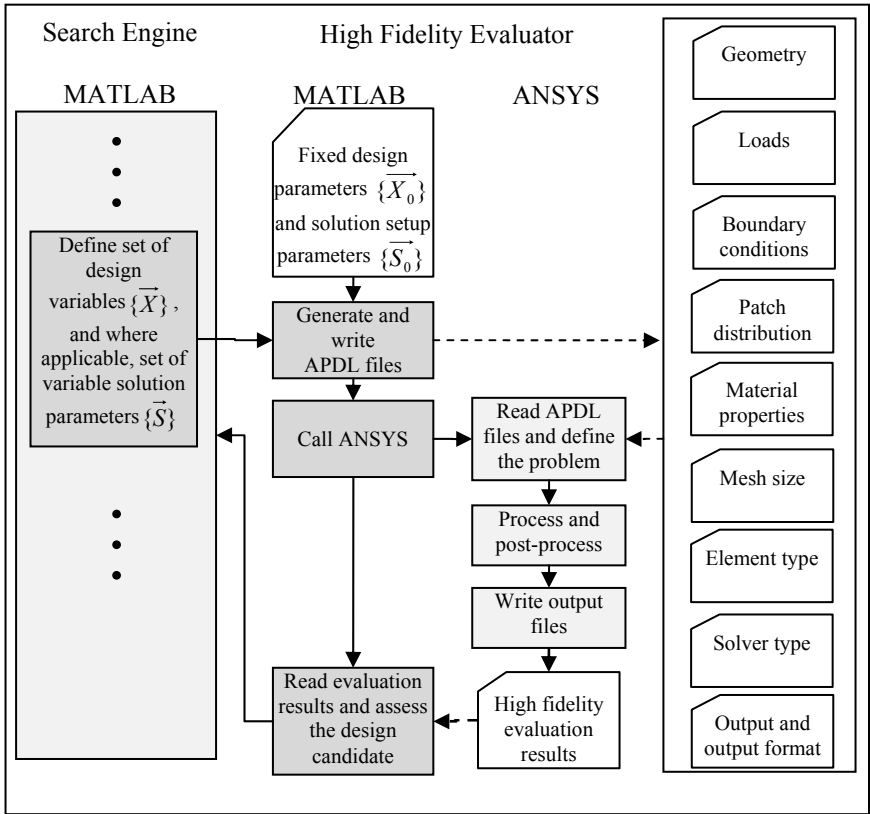


Figure 5: High fidelity analysis via soft-coding.

4.1 Soft coded optimisers for high fidelity design candidate evaluation case study: variable thickness cellular lightweight structures

With the advent of cost effective 3D printing technologies, manufacturing of high performance structures, such as cellular lightweight structures, is undergoing radical changes. Using 3D printers, cells at different locations can be made of different thicknesses, allowing lighter and stiffer structures. Figure 6 shows an aerofoil made of hexagonal cells with different thicknesses optimised for weight subject to stress, stability and deformation constraints. Adopting a

soft-coding approach, optimisation of this structure was carried out by using a genetic algorithm as the search engine written in MATLAB and ANSYS as a high fidelity evaluator.

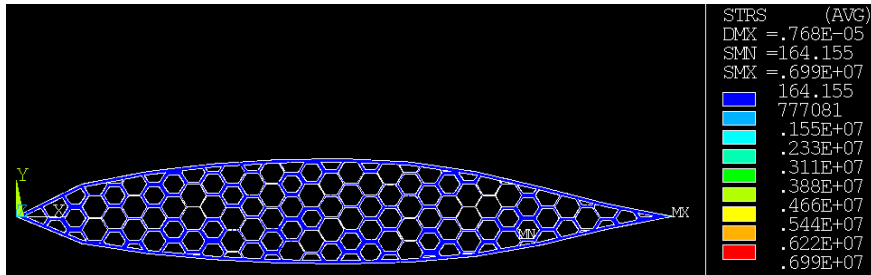


Figure 6: Cellular structure optimization.

Figure 7 shows steps of model generation in MATLAB: (i) defining the outer boundary of each patch and adding patches together (chord length 1 m), (ii) superposition of the aerofoil with a hexagonal cell pattern of a predefined size (2 cm), (iii) defining the inner boundary based on a predefined shell thickness (shown by green), and (iv) assigning a thickness to each cell and defining the inner boundary of each cell based on half of that thickness (the thickness of each side of a cell is defined as the average of the thickness of two adjacent cells). The last step of geometry definition, not shown in this figure, is (v) to find the key points forming each cell as well as those obtained by intersecting the aerofoil inner boundary and cells. Transferring the key points to ANSYS, the geometry of the problem can be fully defined.

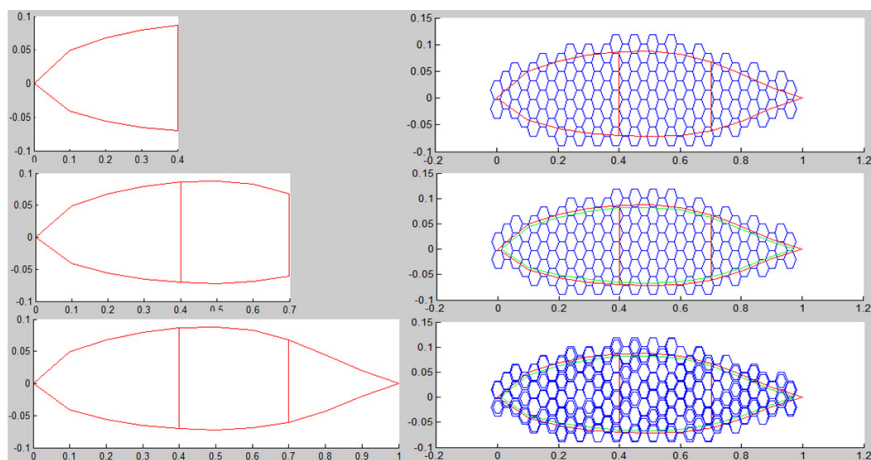


Figure 7: Model generation in MATLAB.

Model generation, updating and applying forces, boundary conditions, material properties, etc, and writing APDL files takes only a fraction of a second in MATLAB environment.

References

- [1] Russo, D., & Rizzi C., Structural optimization strategies to design green products, *Computers in Industry*, 65(3), pp. 470-479, 2014.
- [2] Maheri, A., Noroozi, S. & Vinney, J., Decoupled design of wind turbine adaptive blades, *Renewable Energy*, 32(10), pp. 1753-1767, 2007.
- [3] Maheri, A. & Maheri, M., A robust method for integrated design of trusses supporting distributed loads, *Engineering Structures*, 40, pp. 339-349, 2012.
- [4] Zhou, M. & Yu Wang, M., Engineering feature design for level set based structural optimization, *Computer-Aided Design*, 45(12), pp. 1524-1537, 2013.
- [5] Wang, D., Wu, Z., Fei, Y., & Zhang, W., Structural design employing a sequential approximation optimization approach, *Computers & Structures*, 134, pp. 75-87, 2014.
- [6] Hare, W., Nutini, J. & Tesfamariam, S., A survey of non-gradient optimization methods in structural engineering, *Advances in Engineering Software*, 59, pp. 19-28, 2013.
- [7] Safari, D., Maheri, A. & Maheri, M., Optimum design of steel frames using a multiple-deme GA with improved reproduction operators, *Constructional Steel Research*, 67(8), pp. 1232-1243, 2011.
- [8] Maheri, A., A finite element suite for deformation analysis of composite aeroelastic structures subjected to operational aerodynamic loading, *Proc. IMechE-C, Journal of Mechanical Science*, 226(8), pp. 2062-2076, 2012.
- [9] Maheri, A. & Isikveren, A., Performance prediction of wind turbines utilizing passive smart blades: approaches and evaluation, *Wind Energy*, 13 (2), pp. 255-265, 2010.
- [10] Lagaros, N., Fragiadakis, M., Papadrakakis, M. & Tsompanakis, Y., Structural optimization: A tool for evaluating seismic design procedures, *Engineering Structures*, 28(12), pp. 1623-1633, 2006.
- [11] Clune, R., Connor, J., Ochsendorf, J. & Kelliher, D., An object-oriented architecture for extensible structural design software, *Computers & Structures*, (100-101), pp. 1-17, 2012.

