Contemporary Engineering Sciences, Vol. 10, 2017, no. 21, 1001 - 1010
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ces.2017.79104

Organic-Shaped Structures Design Using Genetic Algorithms and Metaballs

Fernando Martinez

Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Fredy Martinez

Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Cesar Hernández

Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Abstract

Objectives: To develop an algorithm able to generate 3D mechanical structures which look like organic ones. Methods/Analysis: This algorithm starts from a predefined design space and some mechanical constrains (reference points, applied forces, etc.), after, a Finite Element Method (FEM) is used, in order to compute all the resultant stresses into the volume of design space. Then a genetic algorithm uses those results to optimize a 3D structure that accomplishes with the mechanical constrains and a predefined filling percentage of the design space. Findings: The structure proposed is composed by 3D objects named metaballs, which behave like mercury drops, fusing together and creating a rounded and organic-like shape. The fitness function of the genetic algorithm is inspired in the growing of trabecular bone tissues, it means the distribution of metaballs into the volume, is like the one of cells into a bone. Novelty/Improvements: As a result, Organic-shaped structures were automatically obtained according to the starting mechanical constrains.

Keywords: organic-like shape, genetic algorithms, metaballs, finite elements method FEM, bone growing

1. Introduction

Bioinspired algorithms are used in a lot of different areas, including architecture and industrial design [1]–[5], due to most of them, give results that look like organic structures. Specifically, the algorithms of vegetal and animal tissue morphogenesis [6], [7], has been used for creating and optimizing structures. There are different approaches in morphogenetic algorithms, such as: plant grow modeling [8]–[10], which can be supported by using fractals (Lindenmayer systems) [6], [11], [12], cellular growing modeling [13], [14], among others. Another bioinspired algorithm used for designing structures is the modeling and remodeling of bone tissue [15]–[18], which are especially used for optimizing structures under applied forces. The last approach is based on the resultant stresses after applying specific forces to a bone structure under certain constrains; and how it adapts itself and reorganizes its tissue to improve its resistance to the applied forces. Generally, this approach needs to use the Finite Elements Method (FEM) [19]–[22], in order to obtain the stresses matrix needed to adapt the bone tissue. Also, genetic algorithms have been used for designing and optimizing the structure [23]–[26]. The proposal shown in this paper is based on FEM simulations under certain mechanical constrains, of a starting design space; then an genetic algorithm is used for optimizing the position of cells that will compose the final surface. The genetic algorithm uses a minimalist model inspired on bone tissue remodeling to reorganize those cells into design space. Cells into result are represented as metaballs [27], [28] to create the output 3D surface.

2. Methodology

This proposal is composed by several steps, first one is the definition of the design space, the second one is the FEM simulation, the third one is the optimization by means of using the genetic algorithm. These steps will be described in the next subsections.

2.1 Design space

For this implementation, a very simple design space was selected, it is a “prism” of 10x10x20cm (see Fig. 1). This form was selected due to its simplicity and to make easy the result analysis.

Inside the design space is defined a matrix of 2000 cells (metaballs) that will compose the final surface; each cell is separated 1cm for the next one, then a 10x10x20 matrix of metaballs were used.

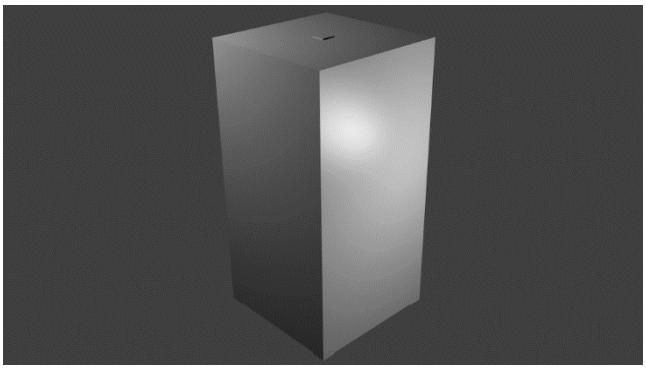

Fig. 1 Definition of design space.

2.2 FEM simulation

Finite Element Method implementation has different parts: first one is to create the finite mesh, second one is to define the constrains (define the physical material, reference points and forces or pressures to apply), the third one is to run the simulation and the last one is the data post-processing. For the meshing process GMSH software was used (see Fig. 2). For the constrain definitions and the simulation itself, CalculiX software was used, a single point force is applied (blue arrow in Fig. 2) and four supports are defined in the lower corners of the structure (orange triangles in fig. 2). Finally the Visualization Tool Kit (VTK) was used for the post-processing, where the inner resultant stresses are shown in color slices (Fig. 3). All the tools used for FEM simulation are freeware and they were implemented by using FreeCAD software.

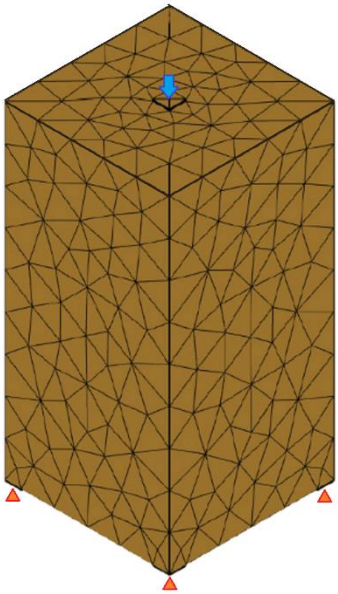

Fig. 2 Design space meshing and constrains definition.

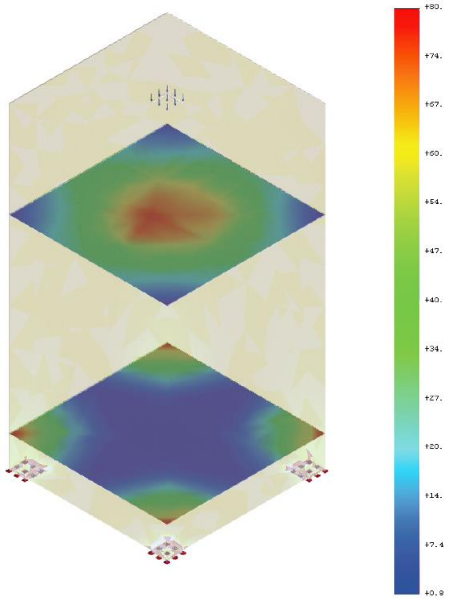

Fig. 3 FEM simulation results (2 slices), it shows the resultant stresses on the solid block, after applying the external force.

For the FEM simulation, the material selected has Polylactic acid (PLA), which is widely used in 3D printing and a lot of other applications. The force was fixed on 1 Newton.

2.3 Genetic Algorithm Definition

A simple genetic algorithm (GA) was used for calculating and optimizing the structure that accomplishes with the initial restrictions, starting from the results of the FEM simulation. This GA has a population of 50 individuals, with simple elitism of 5%, one-point crossing, stochastic uniform selection, crossover fraction of 80% and stochastic uniform mutation of 1% of the population genome. Following section describes the genome definition.

2.3.1 Genome Definition

The genome is defined starting of the data type of the FEM simulation, which brings, for this case, a 3D value matrix of the resultant stresses inside the volume (design space). Depending on the used mesh size, the size of that matrix can be change, that is why, it is necessary to adjust it to the size of matrix cell to be calculated by the GA. For that size adjusting, a decimation process was applied, obtaining a stress matrix S of 10x10x20 positions. The genome of each individual of the population is a string of 2000 Boolean variables, where each one corresponds to each cell or position of the matrix G. In each position of the matrix is a gene in the genome, where “1” means the cell in this position exists and “0” it does not exist. In conclusion, each gen represents the absolute position of a cell in the structure.

2.3.2 Fitness Function

The fitness function was defined trying to emulate the bone tissue growing, using a minimalist model under simple rules. The first characteristic of trabecular structures (porous one inside bones) is the density distribution, this changes along bone depending on the distribution of the stresses in its regular working. This means there will be more cells in the parts of structure under high stress. The second one is the cells connectivity, which means that all of the cell in the structure have to be connected themselves. And the last one is the filling percentage which defines the total density of the bone. Then the fitness function is defined by three factors correspondent to those rules; the first one is the probability factor Fp which is related with the density distribution and is calculated based on the stress matrix S by means of the equation 1. Where l, n and m are de dimensions of cells matrix G, it means 10, 10 and 20 respectively.

$$F_p = \sum_{i=1}^{l}\sum_{j=1}^{n}\sum_{k=1}^{m} \boldsymbol{G}\,(i,j,k) * \boldsymbol{S}(i,j,k) \tag{1}$$

The filling factor Ff defines what percentage of the volume will be filled with cells, based on a previous defined filling percentage pf, as described in equation 2. Where de summation is normalized by the dimensions of the matrix.

$$F_f = \left| p_f - \frac{\sum_{i=1}^{l}\sum_{j=1}^{n}\sum_{k=1}^{m} \boldsymbol{G}\,(i,j,k)}{l * n * m} \right| \tag{2}$$

The connectivity factor Fc define what is the connectivity degree between the cells inside the volume as shown in equation 3. Where N26 is the number of active neighbors of each cell in 3D inside the volume.

$$F_c = \sum_{i=1}^{l}\sum_{j=1}^{n}\sum_{k=1}^{m} \boldsymbol{G}\,(i,j,k)\ * N_{26} \tag{3}$$

Finally, the total fitness function is the product of the factors Fp, Ff y Fc, as shown in equation 4. This fitness function is minimized through the pass of generations in the GA. All the genetic algorithm was developed using MATLAB®.

$$\boldsymbol{f = F_p * F_f * F_c} \tag{4}$$

2.4 Shape Generations by Metaballs

After the evolution of the GA, the resultant cell matrix is represented by means of Metaballs, which are mathematical structures that represent 3D spheres that have

implicit an energy function, which defines what is their surface border. In Metaballs, the closer to the center the intense is the energy function. When two Metaballs are the enough close themselves, the energy function of both, generates only one surface for both, it means those spheres merge as two drops of water under no gravity. Figure 4 shows and example of the effect of the distance over the surface generation (fusion).

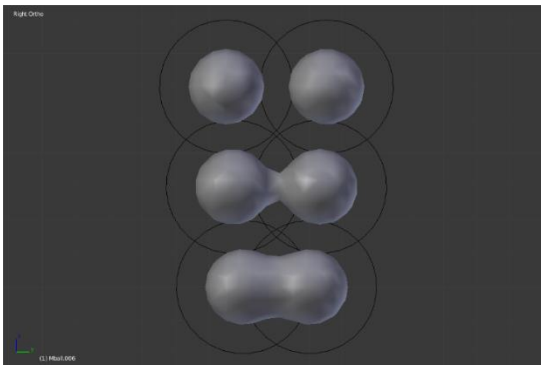

Fig. 4 Fusion of Metaballs depending on the separation distance.

The black line in the external part of each Metaball represents the maximum influence area (minimal value of the energy function). All the resultant shapes obtained when Metaballs are used, are curved, that is why this modeling methodology are commonly used for creating organic-shaped 3D models. Then Metaballs are the most appropriate 3D structures to be used in the context of this research. For the use of Metaballs and the rendering process, Blender software was used.

3. Results

After twenty different experiments the proposed algorithm always generates structures that accomplish the general mechanical constrains, it means they have support for the four base lower points and for the top point where the force was applied. The Figure 5, shows one of the results obtained when defined 30% of filling pf.

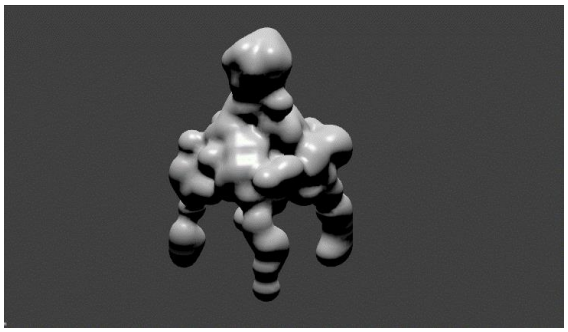

Fig. 5 Resultant structure with 30% of filling.

All of the resultant structure shapes are different, due to the random nature of the GA, but all share the same basic features: they have four slim “Legs”, one “Head” and a central “Body”, which shows that these features respond to the initial mechanical restrictions. The Figure 6 shows another result using 20% of filling.

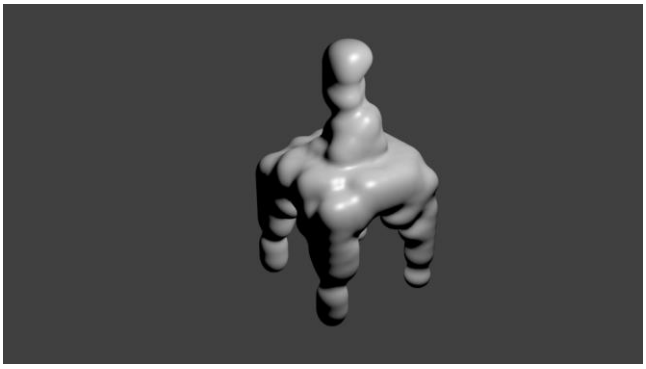

Fig. 6 Resultant structure with 20% of filling.

As shown in figures 5 and 6, the resultant shapes are not symmetric even though the stress profile obtained after the FEM simulation it is. This is due to also the random nature of the GA. If symmetric results are desired, it is possible to divide the original design space in symmetrical components and run the algorithm for one of them.

The number of convergence generations of the different experiments was between 50 and 100, with an average of 67. This shows that the optimization problem as it was proposed is not really complex. All experiments spend an average time of 15 seconds for reach the convergence value (only the GA) running in a regular office PC.

The proposed methodology is able to be used for designing organic-shaped structures, inside simple design spaces, accomplish some mechanical restriction. Despite this methodology uses structural simulations by means of FEM, the main target is the organic design of structures, where the results show its reliability.

4. Future work

It is important to corroborate inside the same algorithm that the obtained shapes accomplish in 100% with the mechanical constrains, simulating them using FEM.

It is possible to join as FEM simulation as genetic algorithm so that the GA uses directly the results of the FEM, doing the process much efficient, due to it is possible to do FEM simulations over the resultant shapes and add them as part of the fitness function.

So far, using the proposed methodology only it is possible to define 3D straight-lines design spaces (cubes, for instance), that is why, it is necessary to formulate an algorithm that automatically fills an arbitrary design space with cells and creates the genome of this cell distribution, in order to be able to effectively apply this algorithm as a design tool.

It is important to explore other possibilities of the GA, such as change elitism, mutation and crossing parameters in order to obtain different results.

It is possible to implement a cell growing algorithm that generates complex structures based few information, and thus to reduce the genome size.

References

[1] A. Pugnale and M. Sassone, Morphogenesis and structural optimization of shell structures with the aid of a genetic algorithm, *Journal-International Assoc. SHELL Spat. Struct.*, **155** (2007), 161.

[2] Y. Meng, H. Guo and Y. Jin, A morphogenetic approach to flexible and robust shape formation for swarm robotic systems, *Rob. Auton. Syst.*, **61** (2013), no. 1, 25–38. https://doi.org/10.1016/j.robot.2012.09.009

[3] T. McGinley, A morphogenetic architecture for intelligent buildings, *Intell. Build. Int.*, **7** (2014), no. 1, 4–15. https://doi.org/10.1080/17508975.2014.970120

[4] A. Menges, Biomimetic design processes in architecture: morphogenetic and evolutionary computational design, *Bioinspir. Biomim.*, **7** (2012), no. 1, 015003. https://doi.org/10.1088/1748-3182/7/1/015003

[5] F. H. M. Sarmiento and A. Delgado, Seres vivos: fuente de inspiración para el diseño artificial, *Rev. Tecnura*, **17** (2013), no. 37, 121–137. https://doi.org/10.14483/udistrital.jour.tecnura.2013.3.a11

[6] C. Fournier and B. Andrieu, ADEL-maize: an L-system based model for the integration of growth processes from the organ to the canopy. Application to regulation of morphogenesis by light availability, *Agronomie*, **19** (1999), no. 3–4, 313–327. ttps://doi.org/10.1051/agro:19990311

[7] H. Kitano, Morphogenesis for evolvable systems, Chapter in *Towards Evolvable Hardware*, Springer, 1996, 99–117. https://doi.org/10.1007/3-540-61093-6_5

[8] P. Prusinkiewicz, Modeling plant growth and development, *Curr. Opin. Plant Biol.*, **7** (2004), no. 1, 79–83. https://doi.org/10.1016/j.pbi.2003.11.007

[9] P. Prusinkiewicz and A.-G. Rolland-Lagan, Modeling plant morphogenesis, *Curr. Opin. Plant Biol.*, **9** (2006), no. 1, 83–88. https://doi.org/10.1016/j.pbi.2005.11.015

[10] B. Porter, A developmental system for organic form synthesis, Chapter in *Australian Conference on Artificial Life*, Springer, 2009, 136–148. https://doi.org/10.1007/978-3-642-10427-5_14

[11] J. Hanan, P. Prusinkiewicz, M. Zalucki and D. Skirvin, Simulation of insect movement with respect to plant architecture and morphogenesis, *Comput. Electron. Agric.*, **35** (2002), no. 2–3, 255–269. https://doi.org/10.1016/s0168-1699(02)00022-4

[12] J. C. Amado, C. E. Garay and F. M. Santa, Implementación de un algoritmo de morfogénesis para el diseño de disipadores de calor, *Tekhnê*, **13** (2016), no. 1, 31–40.

[13] J. R. Wendrich and D. Weijers, The Arabidopsis embryo as a miniature morphogenesis model, *New Phytol.*, **199** (2013), no. 1, 14–25. https://doi.org/10.1111/nph.12267

[14] C. Guoqing, Z. Yan, L. Hui and C. Weixing, Morphogenesis model-based virtual growth system for organs and plant of wheat, *Trans. Chinese Soc. Agric. Eng.*, **2007** (2007), no. 3.

[15] M. Nowak and M. Morzynski, Simulation of trabecular bone adaptation-creating the optimal structure, *Proceedings of the 21st Congress of Theoretical and Applied Mechanics, ICTAM04*, Warsaw, Poland, (2004), 15–21.

[16] M. Nowak, Structural optimization system based on trabecular bone surface adaptation, *Struct. Multidiscip. Optim.*, **32** (2006), no. 3, 241–249. https://doi.org/10.1007/s00158-006-0027-9

[17] R. Ruimerman, Modeling and remodeling in bone tissue. Technische Universiteit Eindhoven Eindhoven, 2005.

[18] P. G. Coelho, P. R. Fernandes, H. C. Rodrigues, J. B. Cardoso, and J. M. Guedes, "Numerical modeling of bone tissue adaptation—a hierarchical approach for bone apparent density and trabecular structure, *J. Biomech.*, **42** (2009), no. 7, 830–837. https://doi.org/10.1016/j.jbiomech.2009.01.020

[19] P. Wei, M. Y. Wang and X. Xing, A study on X-FEM in continuum structural optimization using a level set model, *Computer-Aided Design*, **42** (2010), no. 8, 708–719. https://doi.org/10.1016/j.cad.2009.12.001

[20] P. D. Dunning, H. A. Kim, and G. Mullineux, Investigation and improvement of sensitivity computation using the area-fraction weighted fixed grid FEM and structural optimization, *Finite Elem. Anal. Des.*, **47** (2011), no. 8, 933–941. https://doi.org/10.1016/j.finel.2011.03.006

[21] P. W. Christensen and A. Klarbring, *An Introduction to Structural Optimization*, Vol. 153, Springer Science & Business Media, 2008.

[22] Y. J. Song and E. S. Jeon, Optimal Design of Lightweight Seat Extension Equipment Using Topology Optimization and Design of Experiment, *Int. J. Appl. Eng. Res.*, **12** (2017), no. 9, 1855–1859.

[23] S. M. Shrestha and J. Ghaboussi, Evolution of Optimum Structural Shapes Using Genetic Algorithm, *J. Struct. Eng.*, **124** (1998), no. 11, 1331–1338. https://doi.org/10.1061/(asce)0733-9445(1998)124:11(1331)

[24] S. D. Rajan, "Sizing, Shape, and Topology Design Optimization of Trusses Using Genetic Algorithm, *J. Struct. Eng.*, **121** (1995), no. 10, 1480–1487. https://doi.org/10.1061/(asce)0733-9445(1995)121:10(1480)

[25] K. C. Sarma and H. Adeli, Fuzzy Genetic Algorithm for Optimization of Steel Structures, *J. Struct. Eng.*, **126** (2000), no. 5, 596–604. https://doi.org/10.1061/(asce)0733-9445(2000)126:5(596)

[26] C. K. Soh and J. Yang, Fuzzy Controlled Genetic Algorithm Search for Shape Optimization, *J. Comput. Civ. Eng.*, **10** (1996), no. 2, 143–150. https://doi.org/10.1061/(asce)0887-3801(1996)10:2(143)

[27] X. Jin, Y. Li, and Q. Peng, General constrained deformations based on generalized metaballs, *Comput. Graph.*, **24** (2000), no. 2, 219–231. https://doi.org/10.1016/s0097-8493(99)00156-9

[28] J. Rilling and S. P. Mudur, "On the use of metaballs to visually map source code structures and analysis results onto 3D space, *Ninth Working Conference on Reverse Engineering, 2002. Proceedings*, (2002), 299-308. https://doi.org/10.1109/wcre.2002.1173087

Received: September 21, 2017; Published: November 7, 2017