Structure optimization with a bio-inspired method

J. Miguel Vargas-Felix[1], Salvador Botello-Rionda[1]

[1] Centro de Investigación en Matemáticas. Callejón Jalisco s/n,
Col. Valenciana, Guanajuato, Gto, México 36240
{miguelvargas, botello}@cimat.mx

Abstract. We explore a structure optimization strategy that has an analogous in how bones are formed in embryos, where shape and strengthening are mostly defined. The strategy starts with a rectangular grid of elements of uniform thickness with boundary displacements and force conditions. The thickness of each element can grow or shrink depending on the internal strain, this process is done iteratively. The internal strain is found using the finite element method solving a solid mechanics problem. The final shape depends only on five parameters (von Mises threshold, thickness grow and shrink factors, maximum and minimum thickness). An evolutionary algorithm is used to search an optimal combination of these five parameters that gives a shape that uses the minimal amount of material but also keeps the strain under a maximum threshold. This algorithm requires to test thousands of shapes, thus super-computing is needed. Evaluation of shapes are done in a computer cluster. We will describe algorithms, software implementation and some results.

Keywords. Shape optimization, evolutionary algorithms, finite element.

1 Introduction

Our goal is to create solid structures that work under certain conditions (forces or imposed displacements), while weight, displacement, and strains are minimized.

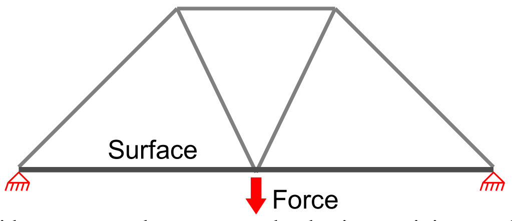


Fig. 1. Bridge structure that support a load using a minimum of material.

To do such, we will apply meta-heuristics with a minimum of assumptions about the problem and its geometry. The evaluation of the structure is done using solid analysis with Finite Element Method [1].

1.1 Topological optimization

When a topological optimization is applied, a domain can be divided in a grid of elements, each element is a degree of freedom (DOF), figure 2 shows an example. A problem with a 2D domain can have thousands of DOF, for 3D problem the number raises to millions. An strategy to reduce the complexity of the search space is to use binary elements.

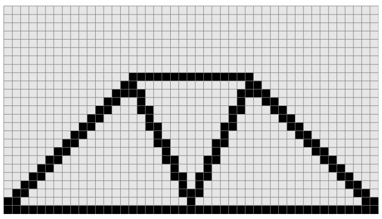


Fig. 2. Example of a grid used for topological optimization.

The aim of the method described below is to work with just a few degrees of freedom, following the idea of how bone shape is defined in mammals.

2 Bone shape

The shape of a bone is mostly defined when embryo is developing. A study [2] explains that at first the bone has a very basic shape, then it grows and adapts itself to have an almost optimal shape to support loads.

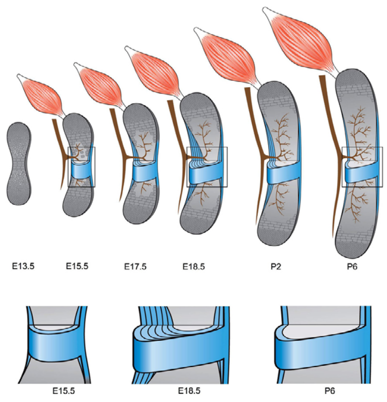


Fig. 3. Model of mouse embryonic bone development, image from [2] (with permission).

In [2] it is demonstrated that the bone reacts to the force created by the growing muscles. Because of external load forces inside the bone strains are generated. In the bone cells where the strain is bigger the ostreoblasts make the concentration of calcium increase, otherwise is reduced. This procedure makes bones to obtain a more resistant shape, with tendency to an optimal.

If the muscles are paralyzed no strain is generated and a bone will not tend to have an optimal shape.

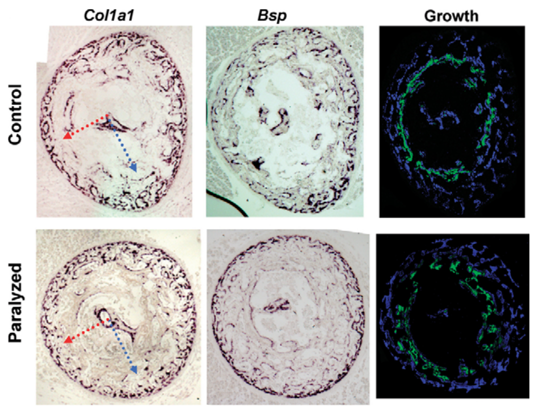


Fig. 4. Osteoblast distribution is controlled by mechanical load, image from [2] (with permission).

3 Structure optimization using internal strain

The Finite element method is used to model the structure, it starts with an empty rectangular domain. The measurement used for internal strain is the von Mises stress or equivalent tensile stress.

Some research have been done on creating simple method that use internal strains to optimize structures [3].

- This method does not use binary elements, instead the thickness of elements variates in a continuous way.
- How thickness will grow or shrink will depend on the von Mises stress inside each element.
- Optimization is done iteratively.
- There is not a fitness function.
- The method works as a cellular automaton.
- There are only five degrees of freedom to control the optimization process.

3.1 Cellular automaton

The rules to control the thickness t_e of the element (cell) are simple:

The thickness can grow by a factor f_{up} or be reduced by a factor f_{down}.

Let σ_{vM} the von Mises strain inside the element and σ^*_{vM} a threshold criteria.

```
if $\sigma_{vM} > \sigma^*_{vM}$ then
    $t_e \leftarrow f_{up} t_e$, with $1 < f_{up}$
```

```
else
    $t_e \leftarrow f_{down} t_e$, with $f_{down} < 1$
```

There are top t_{top} and bottom t_{bottom} limits for the thickness:

```
if $t_e > t_{top}$ then $t_e \leftarrow t_{top}$
if $t_e < t_{bottom}$ then $t_e \leftarrow t_{off}$, where $t_{off} \approx 0.0001$
```

The evolution process of the cellular automaton depends on five parameters:

- von Mises threshold σ^*_{vM}.
- Increase thickness factor f_{up}.
- Reduction of thickness factor f_{down}.
- Top thickness value t_{top}.
- Bottom thickness value t_{bottom}.

3.2 Example: Arc

This is piece of steel with two fixed corners that has to support a force applied on a point, see figure 5.

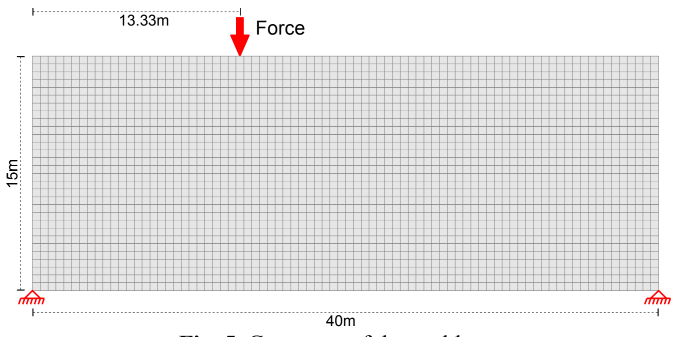


Fig. 5. Geometry of the problem.

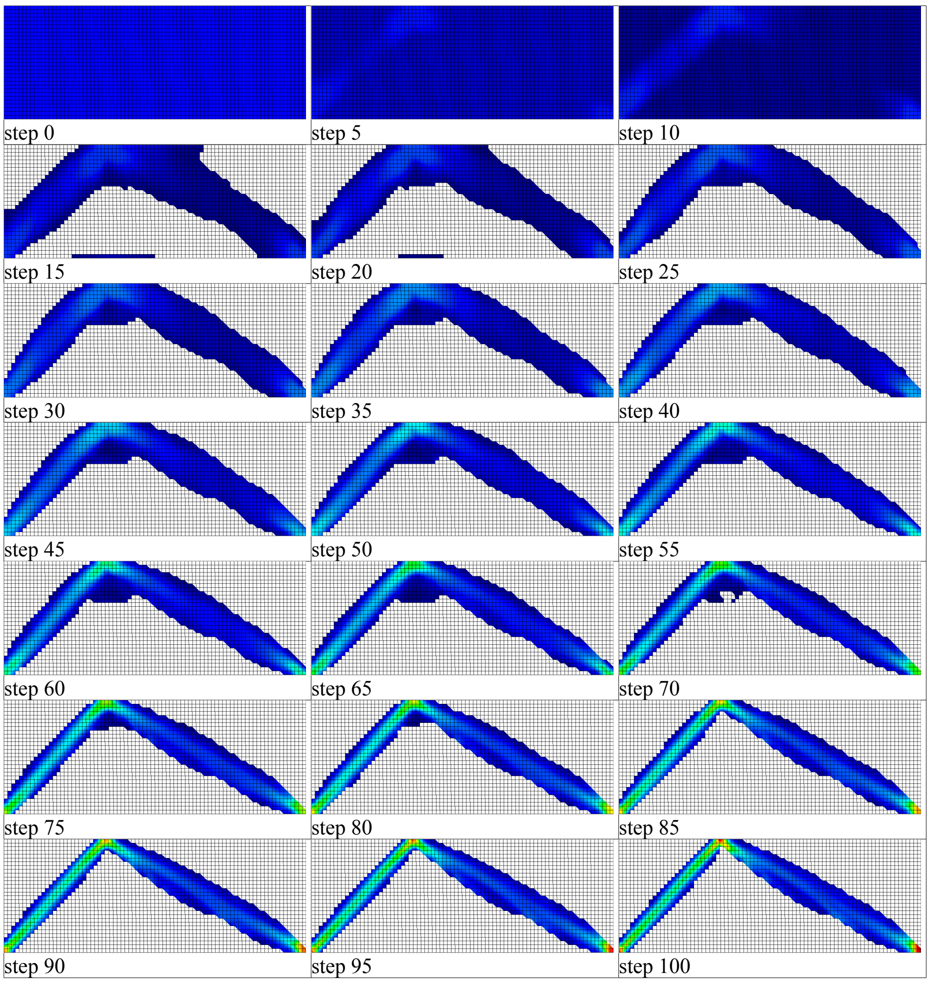
3.2.1 Successful arc structure

For the parameters in table 1, a successful structure is generated.

Table 1. Parameters used to obtain a valid structure.

Parameter	Value
von Mises threshold σ^*_{vM}	2.0
Increase factor f_{up}	1.02
Reduction factor f_{down}	0.91
Top limit f_{top}	8.00
Bottom limit f_{bottom}	0.25

The evolution of the cellular automaton is shown in the sequence of images, warmer colors indicate more thickness, cooler colors less thickness. If thickness falls below f_{bottom} the element is not shown.



The final result is shown in Fig. 6.

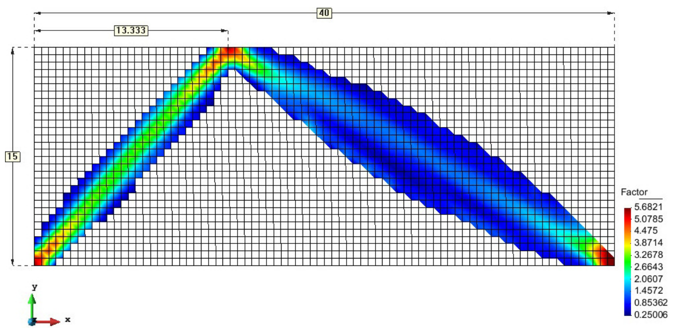


Fig. 6. Final shape.

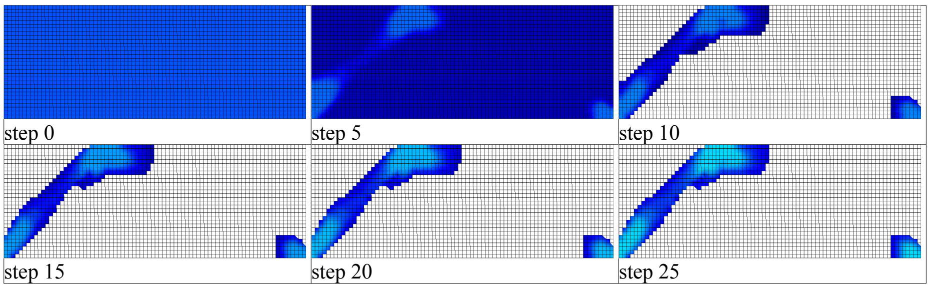
3.2.2 Invalid arc structure

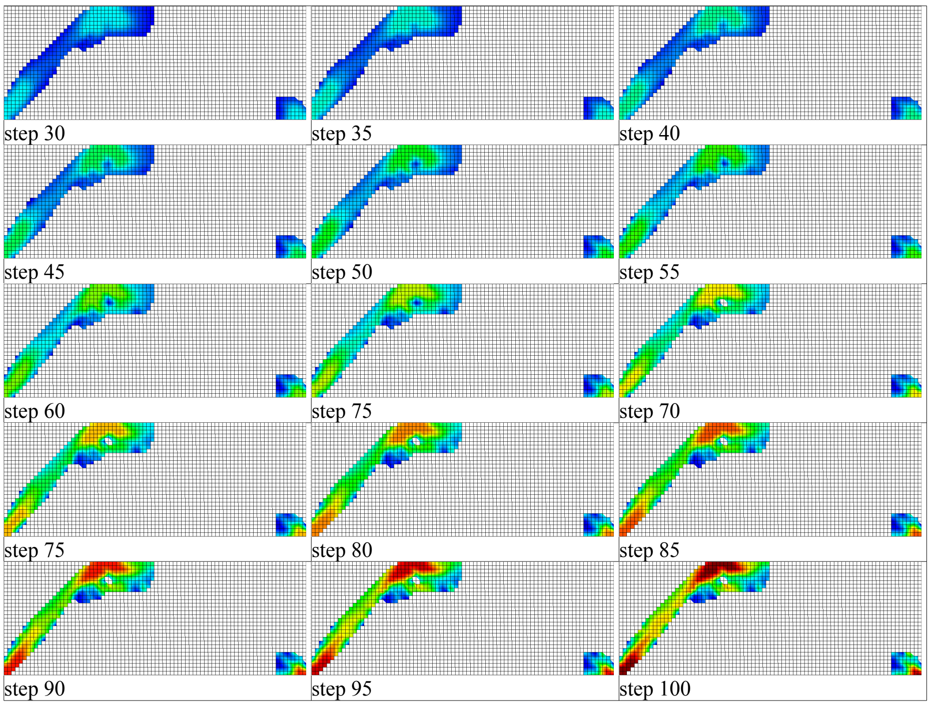
The parameters used are in table 2. In this case the top and bottom limits have been changed. In particular the bottom limit has been increased, with this change the structure fail to success.

Table 2. Parameters used to obtain an invalid structure.

Parameter	Value
von Mises threshold σ^*_{vM}	2.0
Increase factor f_{up}	1.01
Reduction factor f_{down}	0.92
Top limit f_{top}	7.38
Bottom limit f_{bottom}	0.50

Looking at the evolution of the cellular automaton, in both cases success and failure, at early iterations the tickness of the right part of the structure is reduced. In the successful case tickness of this side is increased in the following iterations, until it creates the right part of the structure. If the bottom limit is raised too much the evolution of the right side will be cut too soon, producing an invalid arc structure.





The final result is shown in Fig. 7.

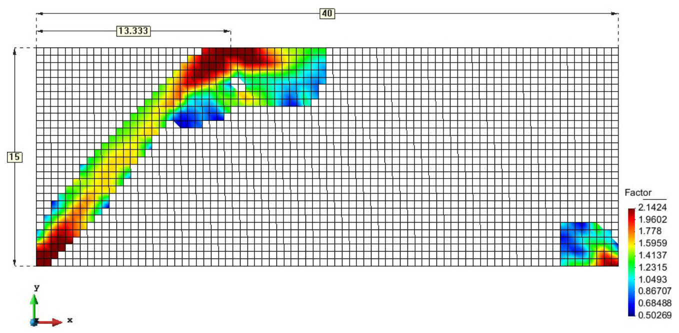


Fig. 7. Final shape.

4 Differential evolution

To search for structures that tend to an optimal a meta-heuristics has to be used. What we have to find are the parameters that improve the structure (in terms of weight, displacement, and internal strain).

The search space will have five dimensions that correspond to the five parameters used to control the cellular automaton.

- von Mises threshold σ^*_{vM}.
- Increase thickness factor f_{up}.
- Reduction of thickness factor f_{down}.
- Top thickness limit t_{top}.
- Bottom thickness limit t_{bottom}.

The search of parameters that produce a good shape is done using differential evolution [4]. The fitness function will measure the weight of the structure w, maximum displacement d and the maximum von Mises in the structure σ_{vM}, we propose a very simple fitness function

$$F \overset{\mathrm{def}}{=} w \cdot d \cdot \sigma_{\mathrm{vM}}.$$

The number of steps of the cellular automaton will be determined heuristically based on some test cases. For our experiments, 100 is the number of steps chosen. Each evaluation of the fitness function will have to complete this amount of steps.

Parameters of the differential evolution will be: population size $N \sim 64$, crossover probability $Cr = 0.8$, and differential weight $D = 0.5$.

The algorithm is of differential evolution is:

```
Let $\mathbf{x}_i \in \mathbb{R}^5$ the $i$-th individual of the population $\mathbf{X} \in \mathbb{R}^{5\times N}$
for each $\mathbf{x}_i \in \mathbf{X}$
    $\mathbf{x}_i^d \leftarrow U(v_{\min}^d, v_{\max}^d),\ d \leftarrow 1,2,\ldots,5$
for $g \leftarrow 1,2,\ldots,g_{\max}$
    for $i \leftarrow 1,2,\ldots,N$
        $a \leftarrow U(1,N),\ b \leftarrow U(1,N),\ c \leftarrow U(1,N)$
        with $i \neq a \neq b \neq c,\ b \neq a,\ c \neq a,\ c \neq b$
        $k \leftarrow U(1,5)$
        for $d \leftarrow 1,2,\ldots,5$
            if $U(0,1) < Cr \ \vee\ d = k$
                $\mathbf{y}_i^d \leftarrow \mathbf{x}_a^d + D \cdot (\mathbf{x}_b^d - \mathbf{x}_c^d)$
            else
                $\mathbf{y}_i^d \leftarrow \mathbf{x}_i^d$
        if $F(\mathbf{x}_i) > F(\mathbf{y}_i)$ then $\mathbf{x}_i \leftarrow \mathbf{y}_i$
        if $F(\mathbf{best}) > F(\mathbf{x}_i)$ then $\mathbf{best} \leftarrow \mathbf{x}_i$
```

5 Implementation

The program to run this method was programed in C++ using the MPI (Message Passing Interface) library for communication between computers in a cluster. The finite element library used to solve each iteration of the cellular automaton was FEMT[1].

1 http://www.cimat.mx/~miguelvargas/FEMT

The cluster used to test this method has 64 cores, to maximize the usage, the population size was chosen to be 64. Solution speed was increased by loading all data for the structure on each core, only the elemental matrix is assembled for each step of the cellular automaton.

The solver used was Cholesky factorization for sparse matrices. Reordering of the matrix is done once and only the Cholesky factors are updated, this calculus is done in parallel using OpenMP [5].

For the examples shown the solution of the finite element problem takes approximately 200ms.

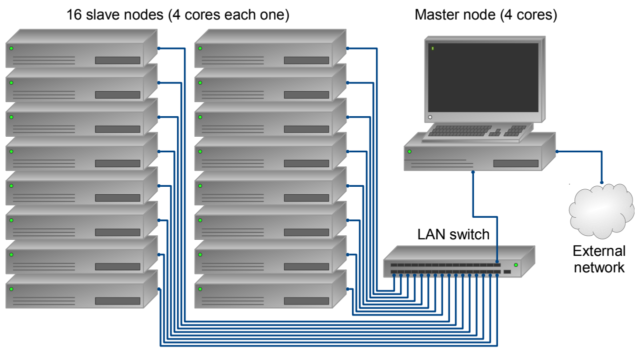


Fig. 8. Diagram of the cluster used to run the optimizer.

The cellular automaton uses 100 iterations, so the calculation of each generation of the differential evolution algorithm takes approx 20s.

6 Global optimization example: Bridge

A steel bar that has two supports on opposite sides, it has to support its own weight and also a force concentrated in the middle.

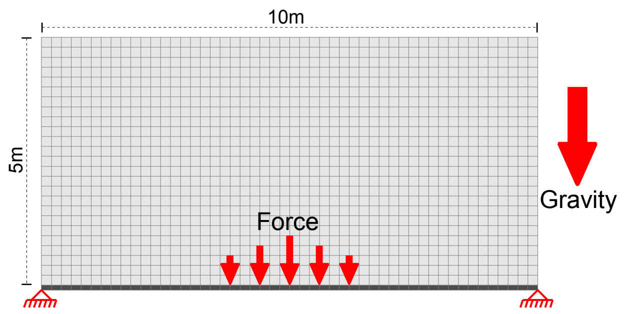
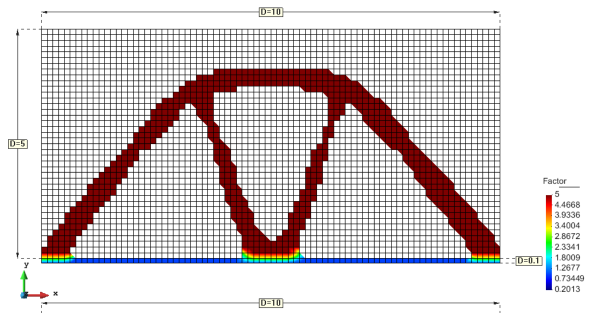


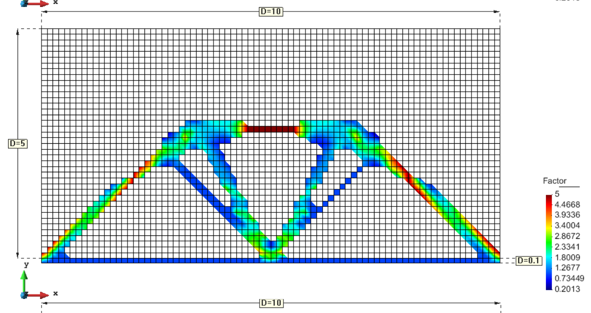
Fig. 9. Geometry of the problem

The next table shows some the best individual after *n* evaluations, an its fitness function.



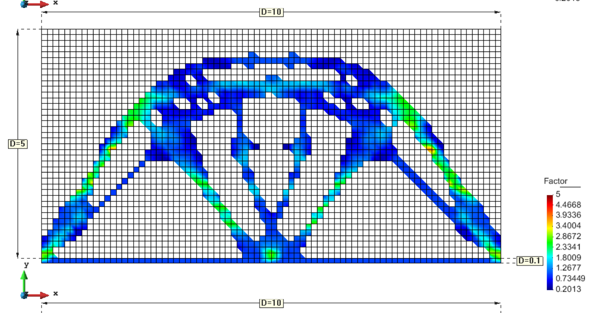
Evaluations: 101

$w = 3.47\times10^{5}$
$d_{\max} = 1.27\times10^{-4}$
$\sigma_{\max} = 1.57\times10^{7}$
$F(\mathbf{x}) = 6.918833\times10^{8}$



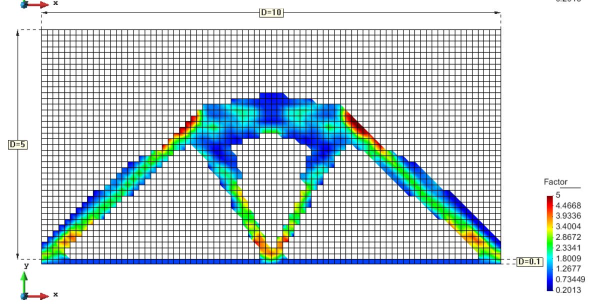
Evaluations: 110

$w = 9.17\times10^{4}$
$d_{\max} = 3.24\times10^{-4}$
$\sigma_{\max} = 1.30\times10^{7}$
$F(\mathbf{x}) = 3.862404\times10^{8}$



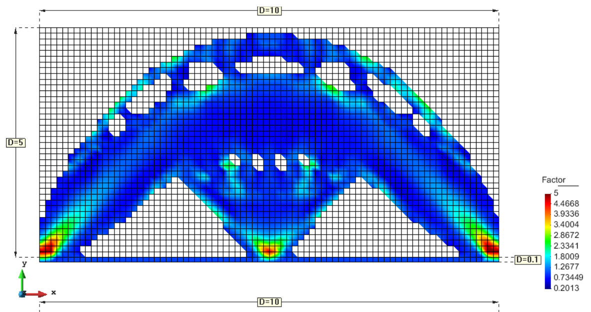
Evaluations: 204

$w = 9.06\times10^{4}$
$d_{\max} = 3.87\times10^{-4}$
$\sigma_{\max} = 1.03\times10^{7}$
$F(\mathbf{x}) = 3.6114066\times10^{8}$



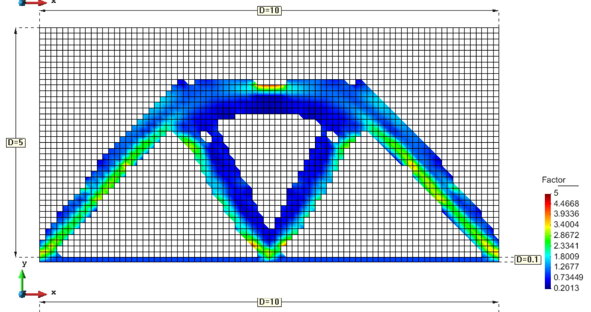
Evaluations: 214

$w = 1.20\times10^{5}$
$d_{\max} = 2.43\times10^{-4}$
$\sigma_{\max} = 1.14\times10^{7}$
$F(\mathbf{x}) = 3.32424\times10^{8}$



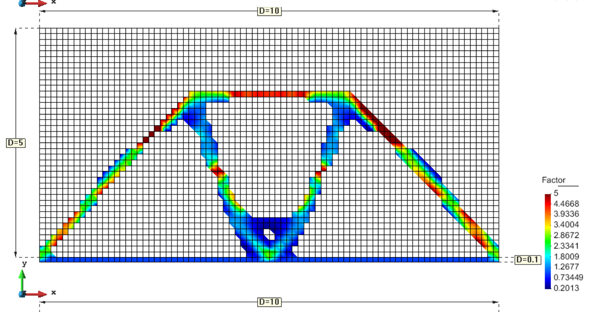
Evaluations: 253

$w = 2.22 \times 10^5$
$d_{\max} = 1.35 \times 10^{-4}$
$\sigma_{\max} = 8.99 \times 10^6$
$F(\mathbf{x}) = 2.694303 \times 10^8$



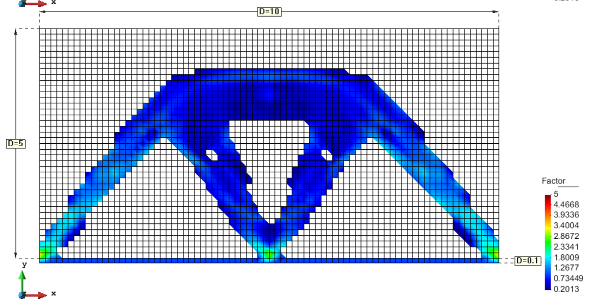
Evaluations: 304

$w = 1.27 \times 10^5$
$d_{\max} = 2.19 \times 10^{-4}$
$\sigma_{\max} = 7.66 \times 10^6$
$F(\mathbf{x}) = 2.1304758 \times 10^8$



Evaluations: 600

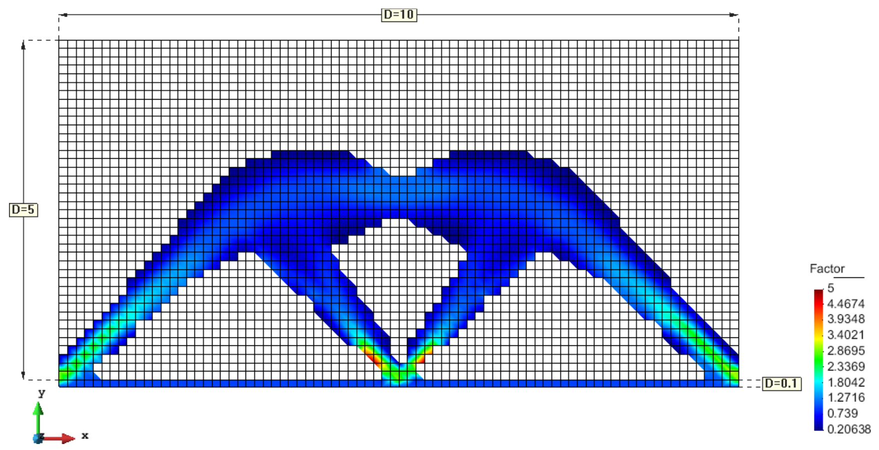
$w = 9.59 \times 10^4$
$d_{\max} = 3.12 \times 10^{-4}$
$\sigma_{\max} = 6.83 \times 10^6$
$F(\mathbf{x}) = 2.04359064 \times 10^8$



Evaluations: 789

$w = 1.00 \times 10^5$
$d_{\max} = 2.73 \times 10^{-4}$
$\sigma_{\max} = 6.75 \times 10^6$
$F(\mathbf{x}) = 1.84275 \times 10^8$

The final structure is:



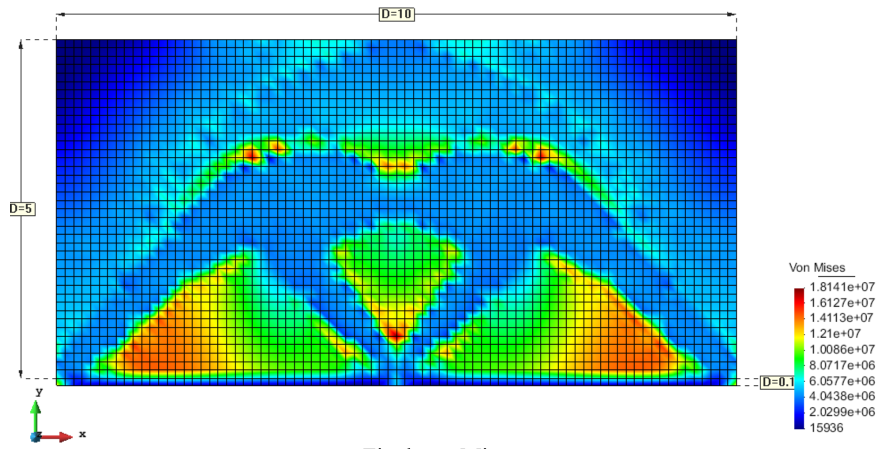
The parameters for the final structure are:

$$\mathbf{x}=\left(\sigma_{vM}^{*}=4.55\times10^{6}, f_{up}=1.03, f_{down}=0.96, f_{top}=5, f_{bottom}=0.2\right)$$

The resulting fitness function:

$$w=1.03\times10^{5}, d_{max}=2.79\times10^{-4}, \sigma_{max}=1.06\times10^{7}$$

$$F(\mathbf{x})=3.046122\times10^{8}$$



Final von Mises.

7 Conclusions

We have presented a bio-inspired method to search for optimal structures under load conditions.

It is interesting to see that in mammals the shape and internal structure of the bone is not codified in the genes. Only some thresholds associated with the behavior of bone cells are codified. With this idea we can reduce an optimization problem with thousands or millions of degrees of freedom (the state of each element in the geometry) to an optimization with just a few degrees of freedom (the parameters used for the cellular automaton).

The evaluation of the fitness functions is expensive because we have to leave the cellular automaton to operate for many steps, we used parallelization in a cluster to overcome this, each core on the cluster evaluates an individual.

Some interesting research can be done in the future, for instance we used a very simple fitness function, a more intelligent selection of this function could be useful to get better and faster results. Also, more complex methods can be used for the optimization, like Estimation of Distribution Algorithms. In the near future we will test this method on 3D structures.

References

1. O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The Finite Element Method: Its Basis and Fundamentals. Sixth edition, 2005.
2. A. Sharir, T. Stern, C. Rot, R. Shahar, E. Zelzer. Muscle force regulates bone shaping for optimal load-bearing capacity during embryo-genesis. Department of Molecular Genetics, Weizmann Institute of Science. Development 138, pp. 3247-3259. 2011.
3. R. Torres-Molina. Un Nuevo Enfoque de Optimización de Estructuras por el Método de los Elementos Finitos. Universitat Politècnica de Catalunya. Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels. 2011.
4. R. Storn, K. Price. Differential Evolution. A Simple and Efficient Heuristic for Global Optimization over Continuous. Journal of Global Optimization Vol. 11, pp. 341–359. 1997.
5. J. M. Vargas-Felix, S. Botello-Rionda. “Parallel Direct Solvers for Finite Element Problems”. Comunicaciones del CIMAT, I-10-08 (CC), 2010.