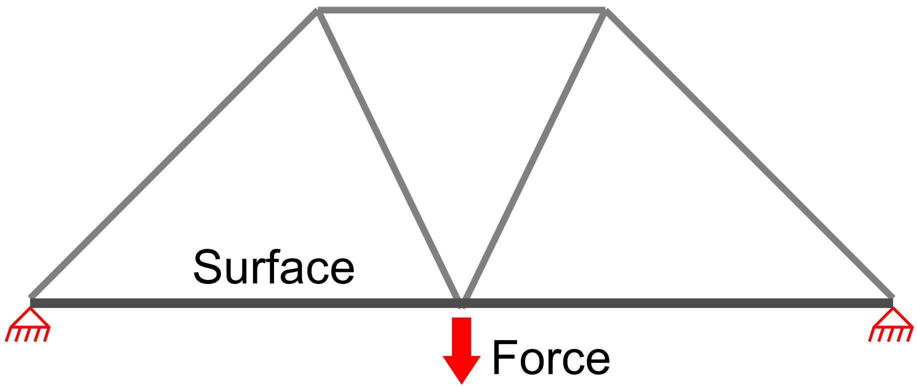
Structure optimization, with a bioinspired method

CIMAT

Miguel Vargas-Félix, Salvador Botello-Rionda
miguelvargas@cimat.mx, botello@cimat.mx

Introduction

Our goal is to create solid structures that are optimal under certain conditions (force, displacement), while the weight, displacement, and strains are minimized.



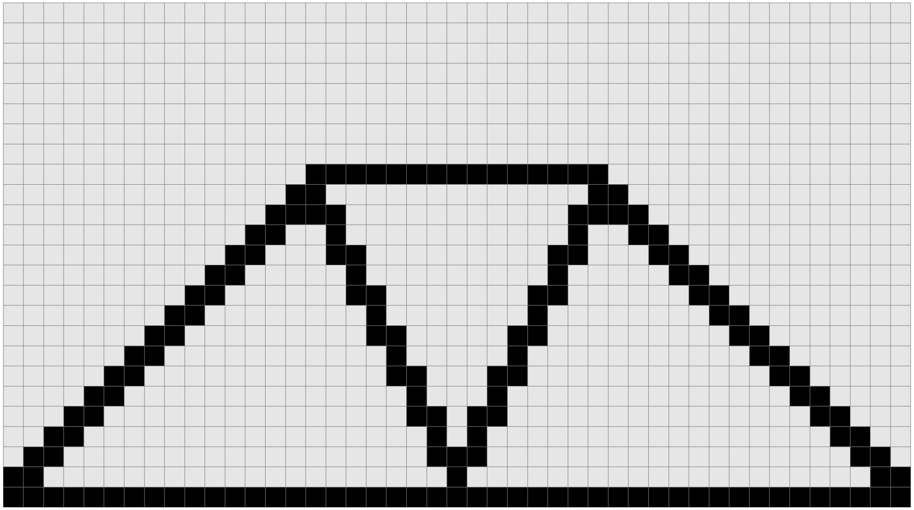
Bridge structure that support a load using a minumum of material.

To do such, we will apply metaheuristics with a minimum of assumptions about the problem an its geometry.

Finite element method is used to modelate the structure, it starts with an empty domain.

Topological optimization

When a topological optimization is applied, a problem can have thousands or millions of degrees of freedom.



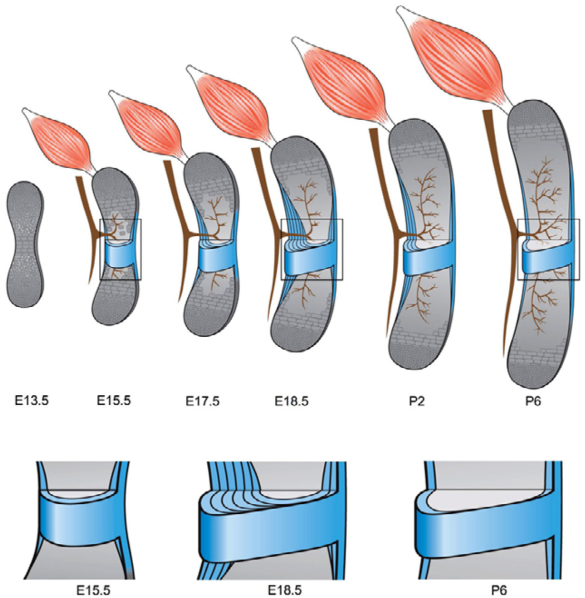
Example of a grid used for topological optimization.

To reduce the search space usually binary elements are used.

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

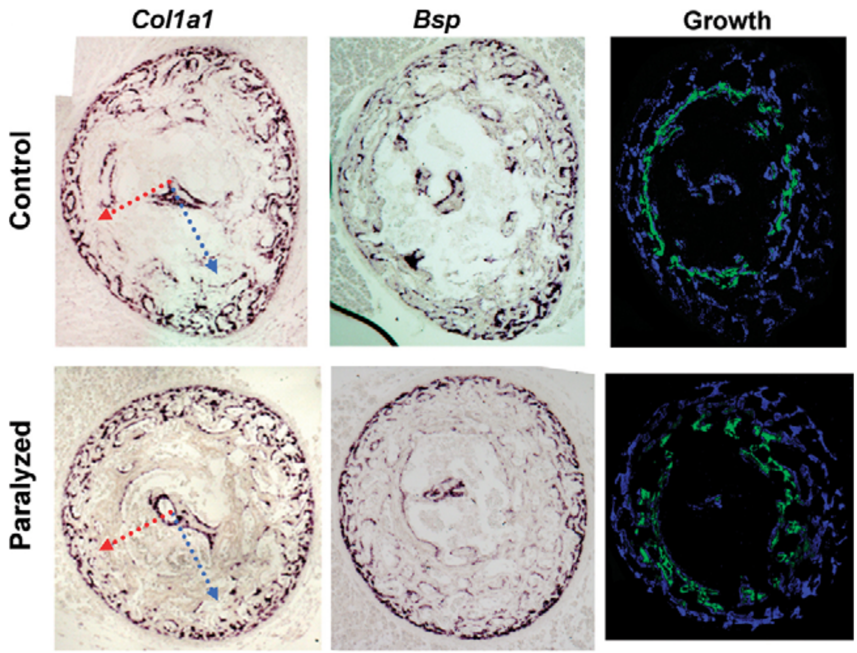
Bone shape

The shape of the bone is defined when embryo is developing. A study [Sharir 2011] explains that at first the bone has a very basic shape, then it grows and adapts itself to have an optimal shape to support loads.



Model of mouse embryonic bone development [Sharir 2011].

In [Sharir 2011] it is demostrated that the bone reacts to the force created by the growing muscles. The strain created inside the bone makes the bone to grow having an optimal shape.



Osteoblast distribution is controlled by mechanical load [Sharir 2011].

If the muscles are paralyzed no strain is generated and the bone never gets an optimal shape.

Structure optimization using internal strain

Some research have been done on creating simple method that use internal strains to optimize structures [Torres 2011].

- This method does not use binary elements, instead the thickness of elements is variated in a continuos way.
- How thickness will grow or shrink will depend on the von Mises inside the element.
- Optimization is done iterativelly.
- 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.

Cellular automaton

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

The tickness 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.

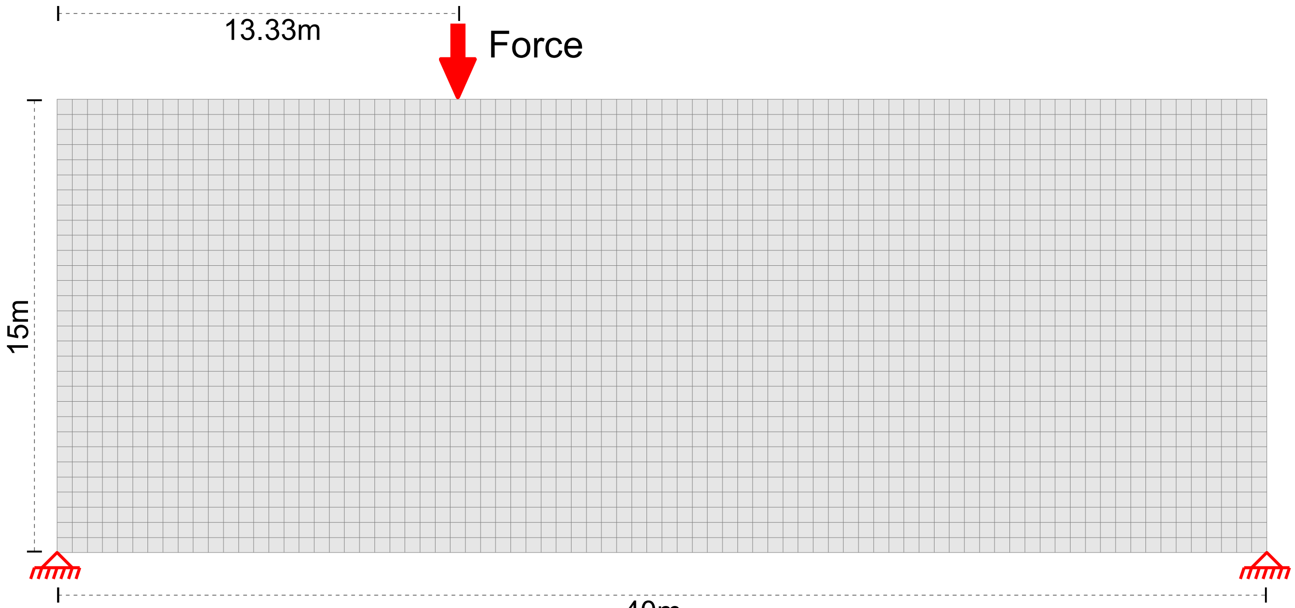
$$
\begin{array}{l}
\text{if } \sigma_{vM} > \sigma^*_{vM} \text{ then} \\
\quad t_e \leftarrow f_{up} t_e, \text{ with } 1 < f_{up} \\
\text{else} \\
\quad t_e \leftarrow f_{down} t_e, \text{ with } f_{down} < 1
\end{array}
$$

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

$$
\begin{array}{l}
\text{if } t_e > t_{top} \text{ then } t_e \leftarrow t_{top} \\
\text{if } t_e < t_{bottom} \text{ then } t_e \leftarrow t_{off}, \text{ where } t_{off} \approx 0.0001
\end{array}
$$

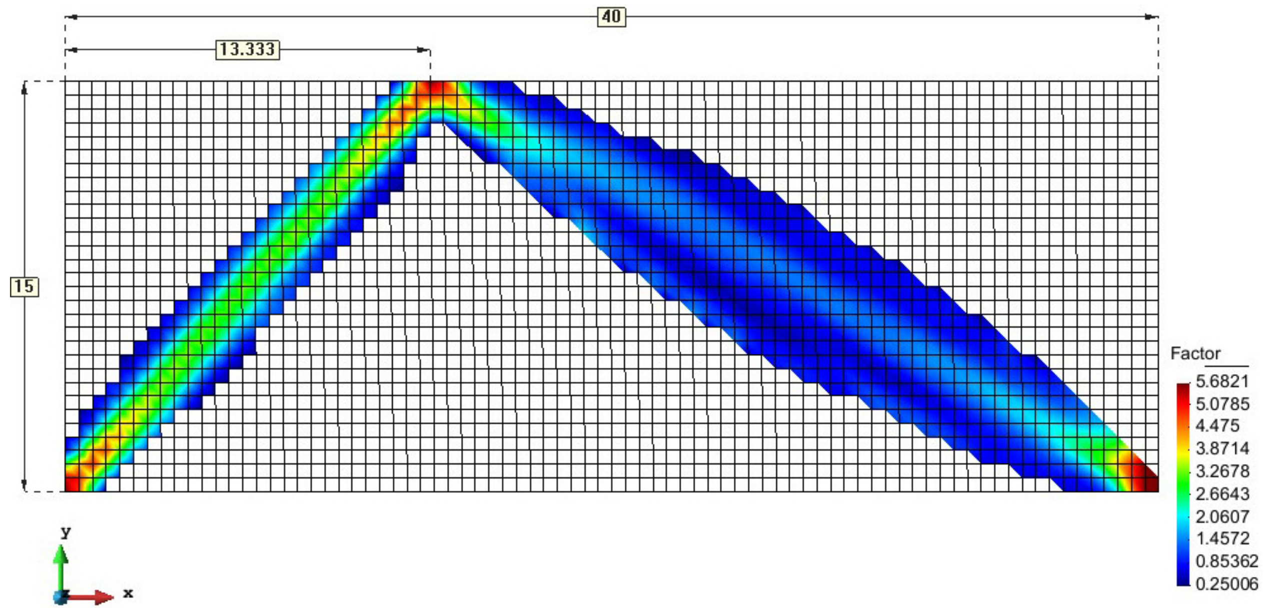
Example: Arc

This is piece of steel with two fixed corners that has to suppor a force applied on a point.



Geometry of the problem.

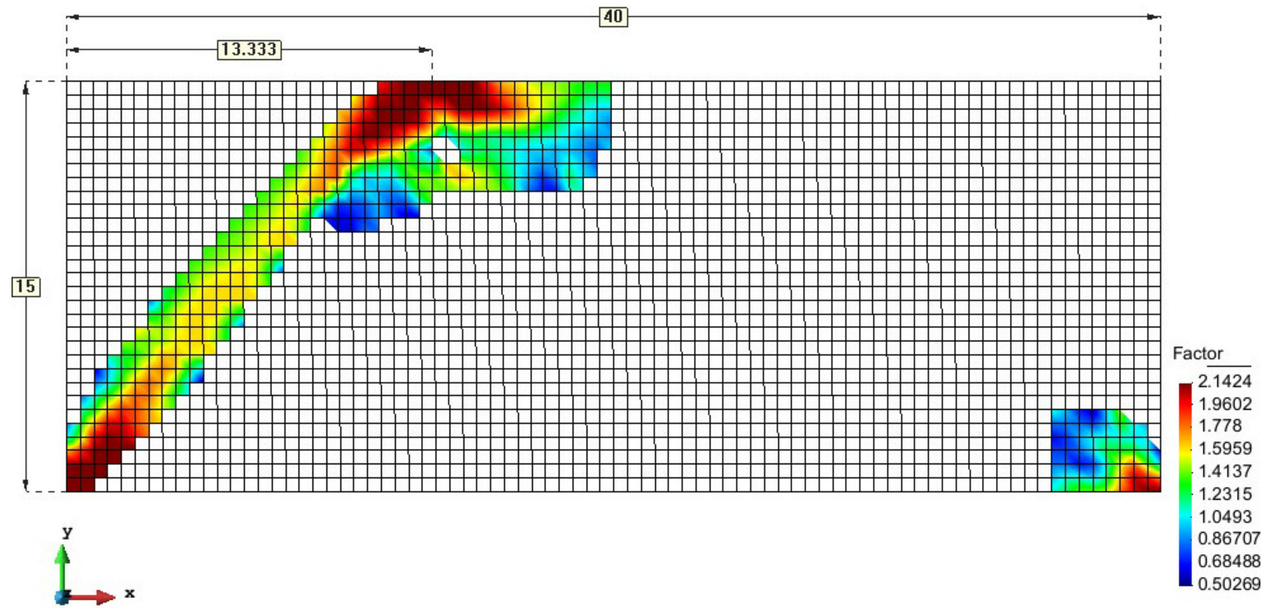
Test: arc.work



arc.work.mpg

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

Test: arc.fail



arc.fail.mpg

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

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

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

To obtain optimal structures a metaheuristics has to be used. The search space will have five dimensions.

Differential evolution

Search of parameters that produce the most optimal shape is done using differential evolution [Storn1997].

The fitness function will measure the weight of the structure w, maximum displacement d and the maximum von Mises in the structure σ_{vM},

$$F \stackrel{\text{def}}{=} w \cdot d \cdot \sigma_{vM}.$$

The number of iterations of the cellular automaton will be determined euristically based on some tests cases.

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:

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 $\mathbf{x}_i \in \mathbf{X}$

$\quad \mathbf{x}_i^d \leftarrow U\left(v_{\min}^d, v_{\max}^d\right),\ d \leftarrow 1,2,\ldots,5$

for $g \leftarrow 1,2,\ldots,g_{\max}$

$\quad$ for $i \leftarrow 1,2,\ldots,N$

$\quad\quad a \leftarrow U(1,N),\ b \leftarrow U(1,N),\ c \leftarrow U(1,N)$

$\quad\quad$ with $i \neq a \neq b \neq c,\ b \neq a,\ c \neq a,\ c \neq b$

$\quad\quad k \leftarrow U(1,5)$

$\quad\quad$ for $d \leftarrow 1,2,\ldots,5$

$\quad\quad\quad$ if $U(0,1) < Cr \ \vee\ d = k$

$\quad\quad\quad\quad \mathbf{y}_i^d \leftarrow \mathbf{x}_a^d + D\cdot\left(\mathbf{x}_b^d - \mathbf{x}_c^d\right)$

$\quad\quad\quad$ else

$\quad\quad\quad\quad \mathbf{y}_i^d \leftarrow \mathbf{x}_i^d$

$\quad\quad$ if $F(\mathbf{x}_i) > F(\mathbf{y}_i)$ then $\mathbf{x}_i \leftarrow \mathbf{y}_i$

$\quad\quad$ if $F(\mathbf{best}) > F(\mathbf{x}_i)$ then $\mathbf{best} \leftarrow \mathbf{x}_i$

Implementation

The optimizator was designed to run in a cluster, each core in the cluster evaluates an individual of the population.

The population size was choosen to be 64.

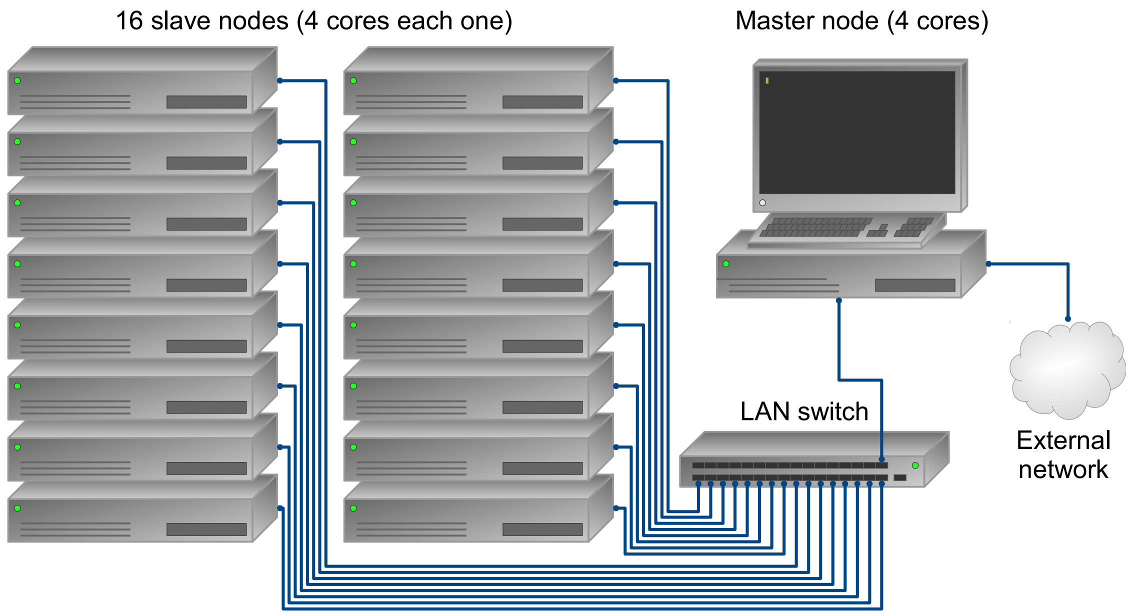


Diagram of the cluster used to run the optimizator.

Solution speed was increased by loading on each core all data for the structure, 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.

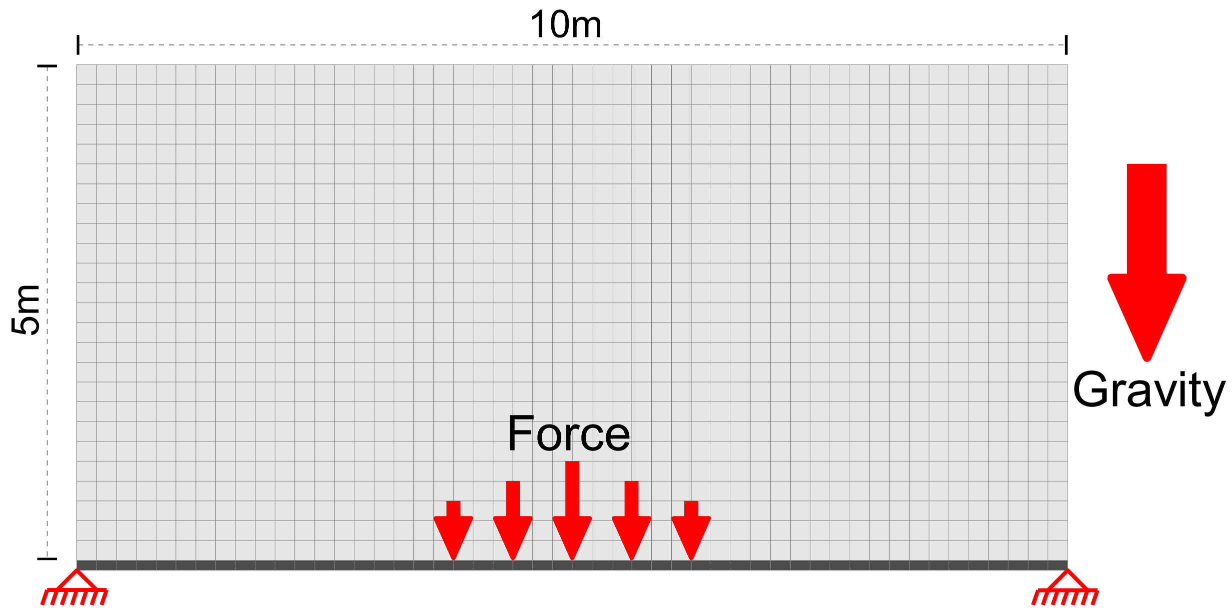
For the examples shown the solution of the finite element problem takes approximately 200ms.

The celular automaton uses 100 iterations.

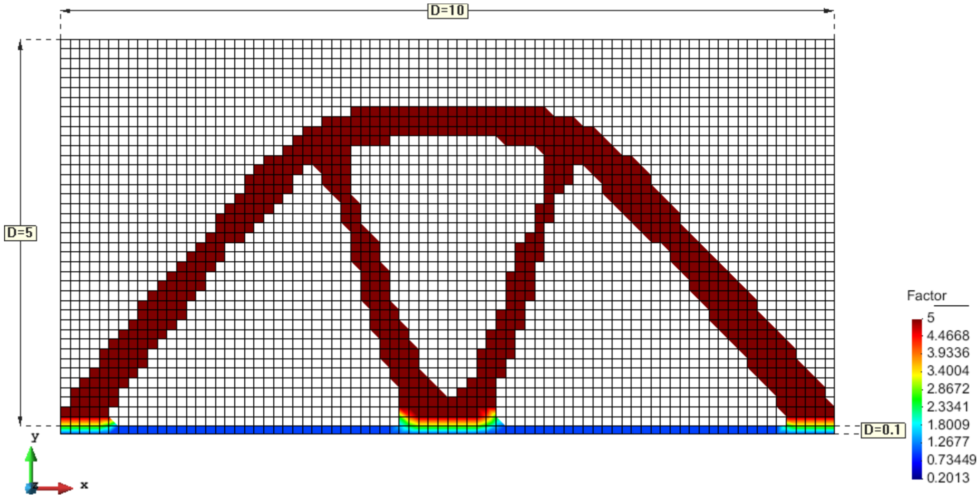
Each generation of the differential evolution algorithm takes approx 20s.

Example: Bridge

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

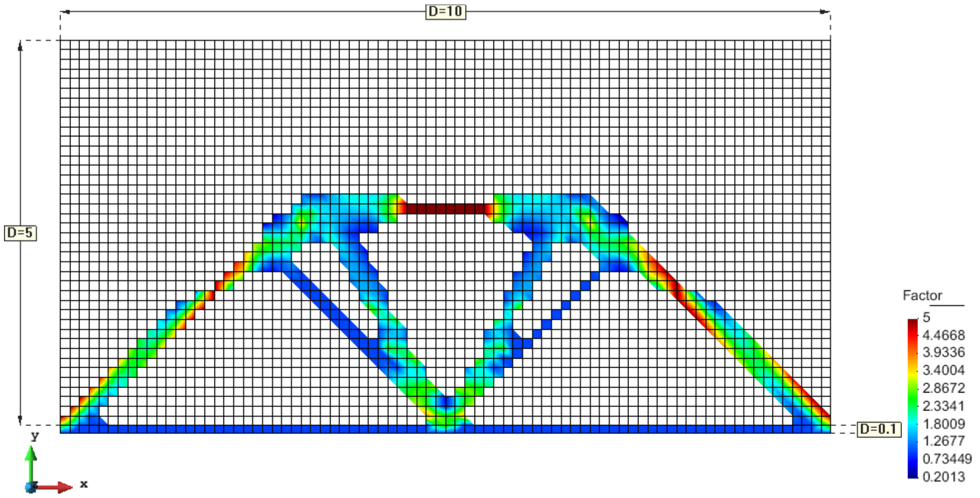


Geometry of the problem



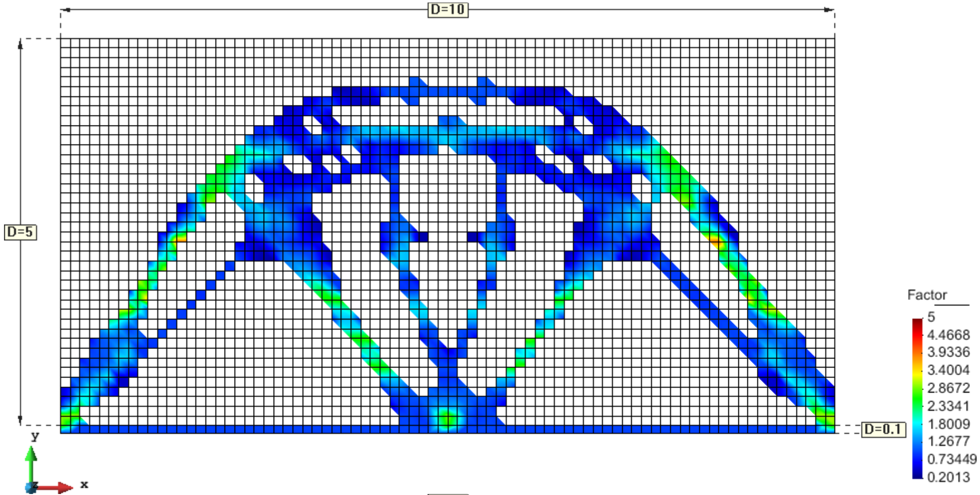
Evaluations: 101

$$w = 3.47 \times 10^{5}$$
$$d_{\max} = 1.27 \times 10^{-4}$$
$$\sigma_{\max} = 1.57 \times 10^{7}$$
$$F(\mathbf{x}) = 6.918833 \times 10^{8}$$



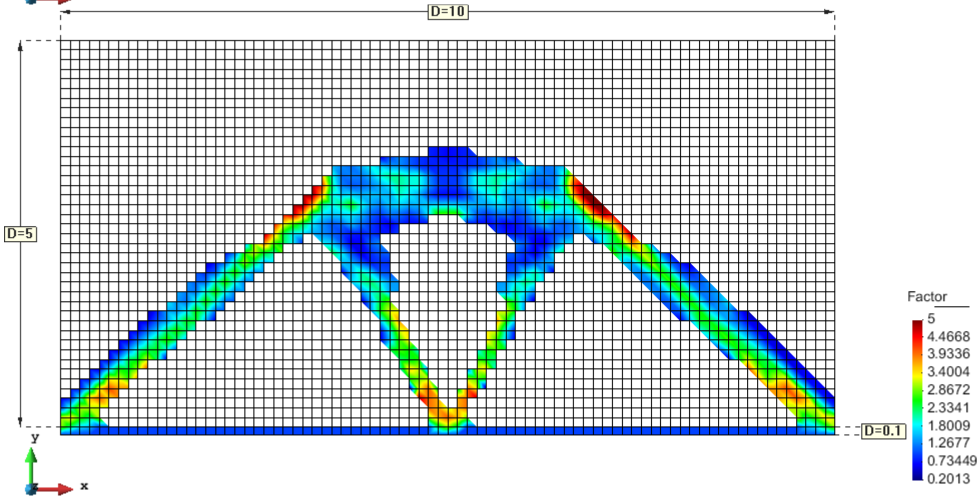
Evaluations: 110

$$w = 9.17 \times 10^{4}$$
$$d_{\max} = 3.24 \times 10^{-4}$$
$$\sigma_{\max} = 1.30 \times 10^{7}$$
$$F(\mathbf{x}) = 3.862404 \times 10^{8}$$



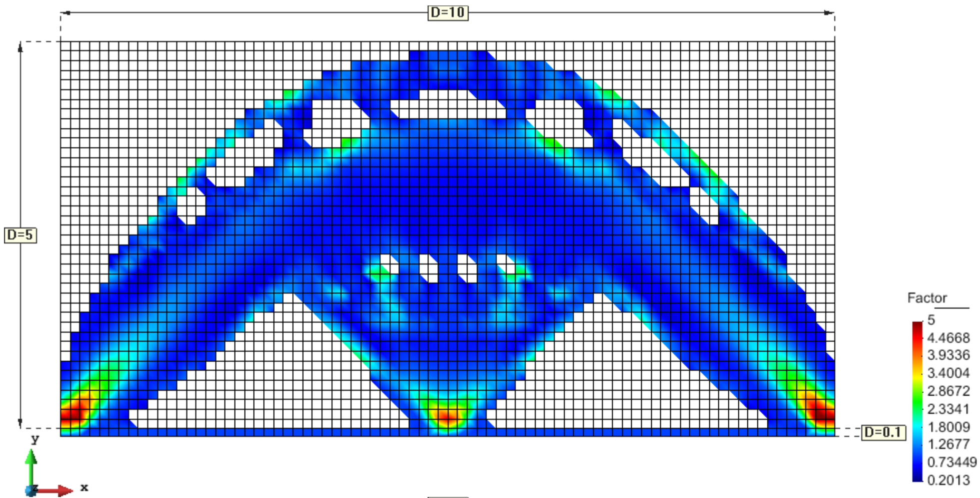
Evaluations: 204

$$w = 9.06 \times 10^{4}$$
$$d_{max} = 3.87 \times 10^{-4}$$
$$\sigma_{max} = 1.03 \times 10^{7}$$
$$F(\mathbf{x}) = 3.6114066 \times 10^{8}$$



Evaluations: 214

$$w = 1.20 \times 10^{5}$$
$$d_{max} = 2.43 \times 10^{-4}$$
$$\sigma_{max} = 1.14 \times 10^{7}$$
$$F(\mathbf{x}) = 3.32424 \times 10^{8}$$



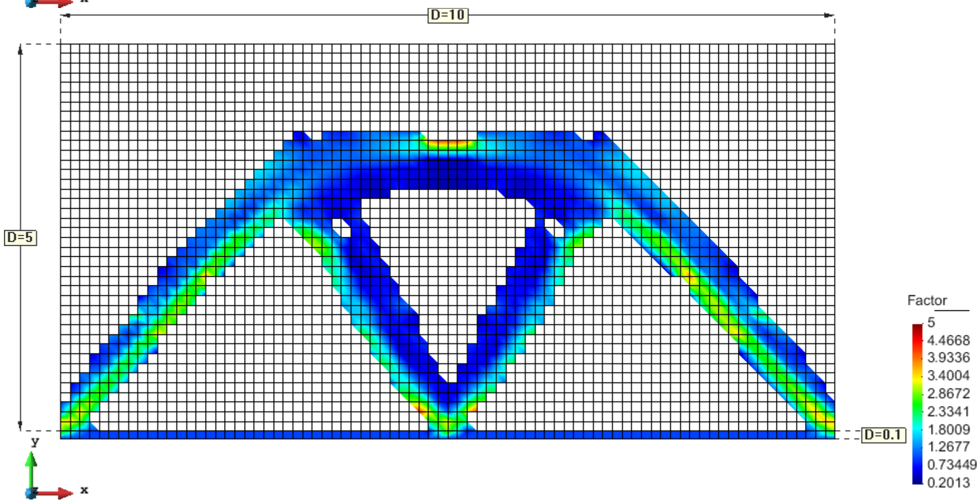
Evaluations: 253

$$w = 2.22 \times 10^{5}$$

$$d_{\max} = 1.35 \times 10^{-4}$$

$$\sigma_{\max} = 8.99 \times 10^{6}$$

$$F(\mathbf{x}) = 2.694303 \times 10^{8}$$



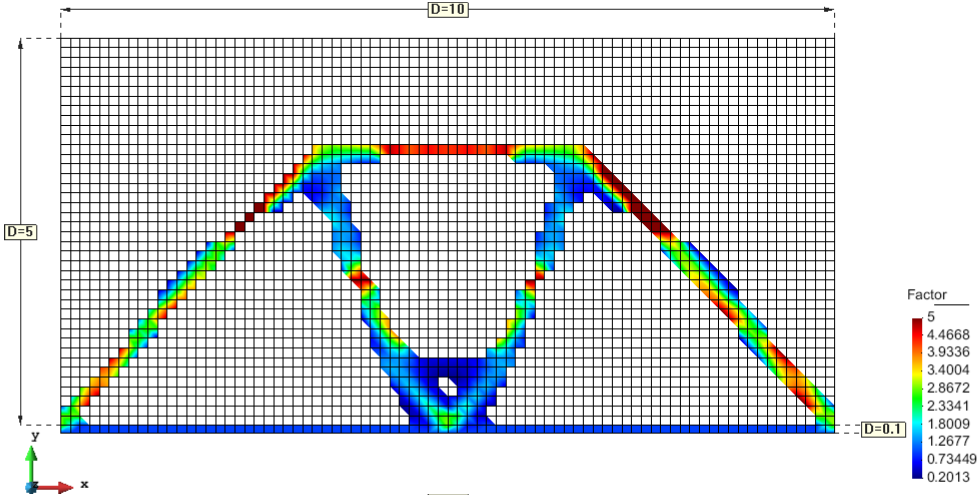
Evaluations: 304

$$w = 1.27 \times 10^{5}$$

$$d_{\max} = 2.19 \times 10^{-4}$$

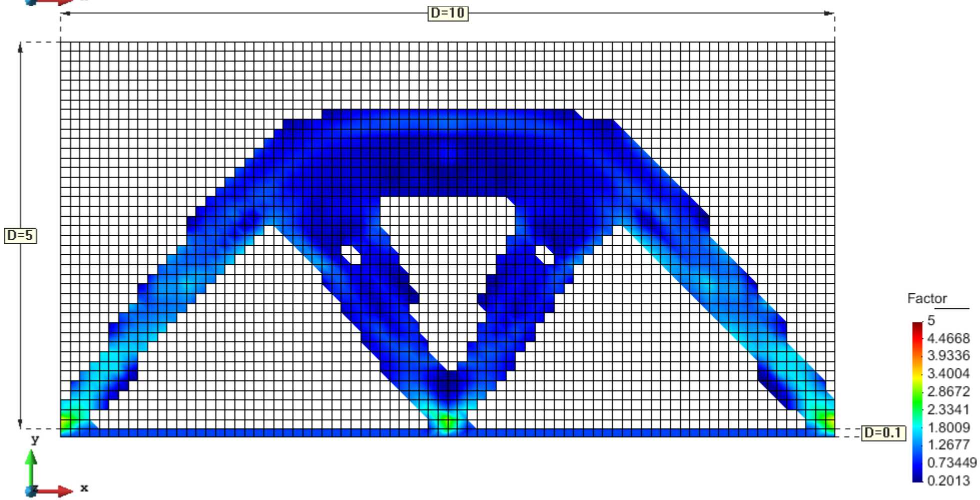
$$\sigma_{\max} = 7.66 \times 10^{6}$$

$$F(\mathbf{x}) = 2.1304758 \times 10^{8}$$



Evaluations: 600

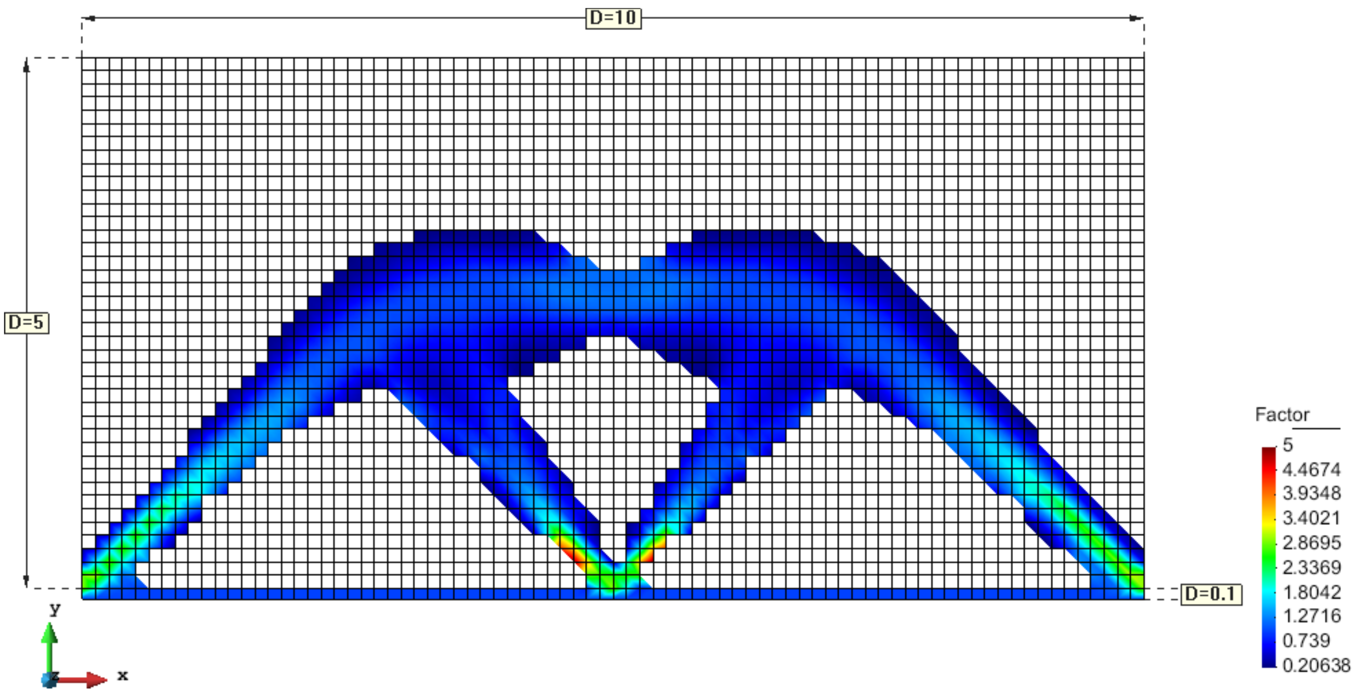
$$w = 9.59 \times 10^{4}$$
$$d_{\max} = 3.12 \times 10^{-4}$$
$$\sigma_{\max} = 6.83 \times 10^{6}$$
$$F(\mathbf{x}) = 2.04359064 \times 10^{8}$$



Evaluations: 789

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

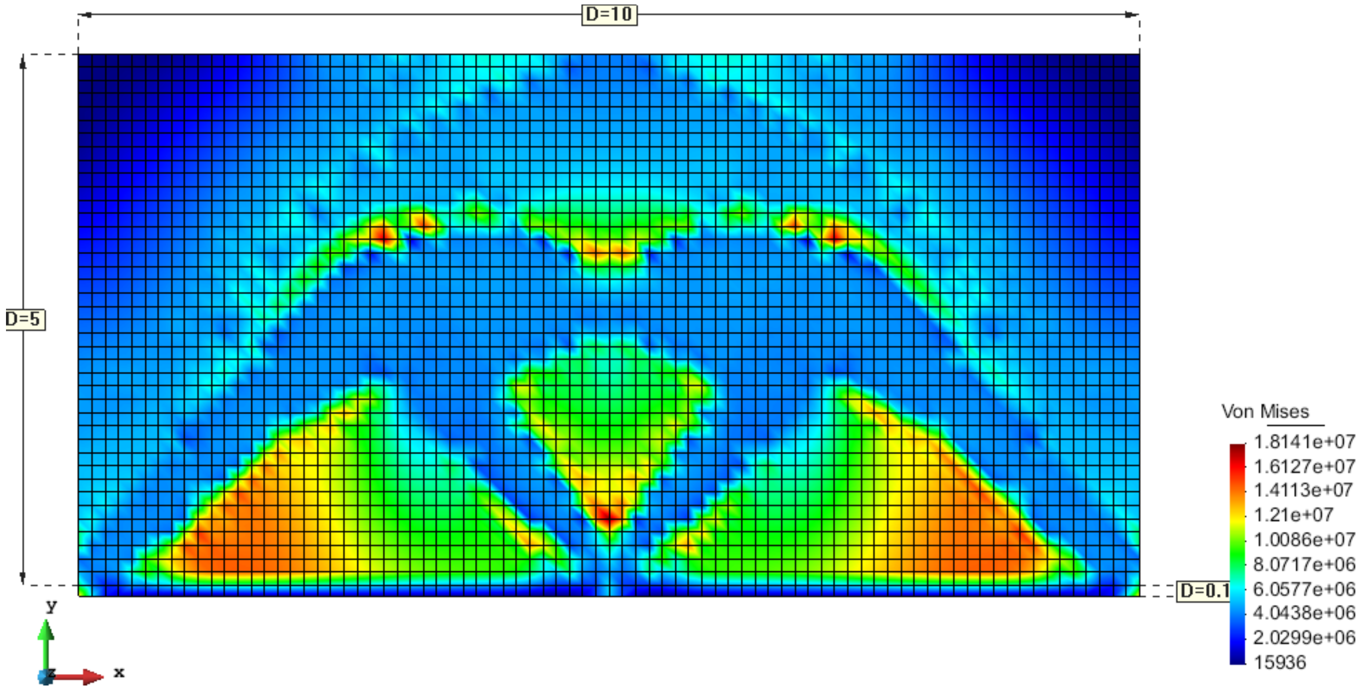
Test: bridge.factor



bridge.factor.mpg

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

Test: bridge.von_mises



bridge.von_mises.mpg

$$w = 1.03 \times 10^{5}, d_{\max} = 2.79 \times 10^{-4}, \sigma_{\max} = 1.06 \times 10^{7}$$

$$F(\mathbf{x}) = 3.046122 \times 10^{8}$$

Conclusions

We have presented a bio-inspired method to obtain 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 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 computer 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 EDAs.

In the near future we would like to test this method on 3D structures.

References

[Sharir 2011] A. Sharir, T. Stern, C. Rot, R. Shahar, E. Zelzer. Muscle force regulates bone shaping for optimal load-bearing capacity during embryogenesis. Department of Molecular Genetics, Weizmann Institute of Science. Development 138, pp. 3247-3259. 2011.

[Torres 2011] 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.

[Storn1997] 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.