A PENALTY BASED GENETIC ALGORITHM FOR MESH OPTIMIZATION OF 2D NON-CONVEX POLYGONS

Olivier R. Gouveia[1,2,†], José Miranda Guedes[2], Rui B. Ruben[1]

[1]*Center for Rapid and Sustainable Product Development (CDRSP) - ESTG, Polytechnic of Leiria, Marinha Grande, Portugal*
[2]*Institute of Mechanical Engineering (IDMEC), Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal*
[†]*Corresponding author – olivier.gouveia@ipleiria.pt*

ABSTRACT

Mesh quality is a crucial aspect when dealing with partial differential equations (PDEs) as it affects both solution accuracy and respective solving time. Optimization of 2D complex domains with non-convex boundary can be difficult to perform using classical gradient-based optimization methods with boundary inequality constrains as they fail to properly define the non-convex search space of the design variables. The proposed procedure uses an evolutionary optimization method, a genetic algorithm (GA), with a penalty-based cost function to obtain the optimal coordinates for inner nodes of generic non-convex polygons meshes with triangular elements.

Keywords: mesh generation, mesh optimization, genetic algorithm, non-convex polygon

1. INTRODUCTION

Numerical methods such as the Finite Element Method (FEM) are commonly used for solving Partial Differential Equations (PDE's). The capacity of a numerical PDE simulation to correctly represent the underlying physics is intrinsically linked, amongst other factors, to the domain's discretization, called mesh. In fact, the accuracy, stability, and efficiency of the numerical methods are directly impacted by the quality of the elements of the mesh. Low quality elements can lead to slow convergence rate, inaccurate results, and consequent problem misinterpretation. Optimization of unstructured meshes is challenging and numerical optimization methods often experience problems such as slow convergence and getting stuck in a local minima which can be far from the optimal nodal positions. The objective function as well as the design variable vector are then function of all these coordinates. While it is possible to iteratively compute nodal position updates to this consequently large vector, it is quite inefficient to do so, [1].

The use of gradient based methods for mesh optimization problems can lead to potential issues such as increased computational overhead for computation of second-order derivative information for the Hessian matrix of the objective function and the possibility of the Hessian matrix not being positive definite, [1]. Hence, derivative-based line-search solvers cannot be directly applied to solve these mesh optimization problems, [2]. On the other, even if the nodal connectivity is known a-priori and enforced throughout the optimization procedure, the definition of an analytical expression of the overall quality as function of the nodal coordinates of the inner nodes quickly becomes impractical and complex to obtain as well as its derivatives. Consequently, derivative-free optimization methods become good candidates. Resulting from their global optimization property, genetic algorithms (GA) have been widely applied in different relevant fields of engineering such as machine learning or nonlinear optimization. These methods proved to have an efficient and robust heuristic for optimal search in complex spaces solving complex optimization, [3].

To optimize a given mesh, one must ensure that the interior nodes remain within the domain's boundary. For convex domains, this task can be easily performed by assigning an inequality constraint for each segment of the discretized boundary. However, this method fails to properly define non-convex domains. To overcome this, a penalty-based objective function is proposed to force the fitness landscape to converge its optimal value within the desired boundary. This research note then focuses on the development of a mesh optimization procedure for non-convex domains by mean of a GA using penalty-based objective function.

2. PROBLEM FORMULATION

We define a conforming triangular partition $\overline{\Omega}$ of an arbitrary bounded domain Ω, as the union of sets of triangular subdomains (or elements) $\mathcal{T} = \{T_1, \ldots, T_N\}$.
The set of nodes of $\mathcal{T}$ can be split into the set of boundary and interior nodes, $\mathcal{N}_B$ and $\mathcal{N}_I$, respectively.

2.1 Non-Convex domain definition

To optimize a triangular partition, boundary nodes $\mathcal{N}_{\boldsymbol{B}}$ can be fixed and the interior nodes $\mathcal{N}_{\boldsymbol{I}}$ relocated to optimize a given mesh metric. For convex domains, the use of boundary inequality constraints can be applied to restrict the feasible space within the discretized boundary, as $\overline{\Omega}$ can be defined as the intersection of the resultant half-spaces, [4]However, for non-convex domain, the use of such mathematical and geometrical property fails as the intersection of the half-spaces defined by the inequalities of the domains' boundary fail to correctly represent the full geometry. To unsure convergence into the desired domain, a penalty-based objective function was implemented into the optimization process, as explained further in section 2.4.

2.2 Mesh Evaluation Metric

Many mesh quality metrics can be found in the literature such as element skewness, Jacobian condition number, orthogonal quality, amongst others.
Area is orientation, skew, and aspect ratio invariant but depends on the scale, [5]. Thus, this metric is not bounded between 0 and 1. However, by applying statistical operators, such as variance, to the area distribution of elements of a given mesh, one can obtain a metric from 0 to ∞ that presents better information about the mesh uniformity
When a fixed number of nodes is considered, r-adaptivity allows the nodes to move within the domain to uniformly distribute the value of the chosen quality metric, [6]. However, r-adaptivity implies an unchanged nodal connectivity between nodes. For connectivity changing procedures with an unchanged number of nodes, an adapted h-adaptivity procedure can be performed. By varying the location of the nodes and, consequently, the area of mesh elements, a possible metric to evaluate element area uniformity can be defined as:

$$Q = \sigma^2(\mathbf{A}) \tag{1}$$

In which Q is the variance σ^2 of the element areas throughout the mesh given by the vector $\mathbf{A}$. By creating equal area elements throughout the mesh, hence better uniformity, the metric Q will tend to 0.

2.3 Objective Function

As presented, the objective of the optimization process will be to minimize the element area variation Q within the desired bounded partition $\bar{\Omega}$. This will be achieved through an optimal positioning of the interior nodes $\mathcal{N}_{\boldsymbol{I}}$ with nodal coordinates $\{\boldsymbol{x}, \boldsymbol{y}\}$. Hence, the mathematical formulation of the optimization problem is stated as follows:

$$\begin{aligned} &\min Q \\ &st. (\boldsymbol{x}, \boldsymbol{y}) \in \bar{\Omega} \end{aligned} \tag{2}$$

Genetic algorithms, however, often require the definition of upper and lower bounds for the design variables. Since each node will be required to "explore" the entire domain, these bounds form a quadrilateral search space ψ that inscribes the domain $\bar{\Omega}$, as depicted in Figure 1.
As a result, the original constrained formulation was transformed into an unconstrained problem by adding a geometrical penalty term. The new unconstrained optimization formulation can then be defined as minimizing the penalized function $F(\mathcal{N}_{\boldsymbol{I}}, r)$:

$$F(\mathcal{N}_{\boldsymbol{I}}, r) = f(\mathcal{N}_{\boldsymbol{I}}) + P(\mathcal{N}_{\boldsymbol{I}}, r) \tag{3}$$

In which

$$f(\mathcal{N}_{\boldsymbol{I}}) = \sigma^2(\mathbf{A}) \tag{4}$$

Where $\mathbf{A}$ is an array containing the element areas of the resulting mesh. The penalty term $P(\mathcal{N}_{\boldsymbol{I}}, r)$ is defined as such:

$$P(\mathcal{N}_{\boldsymbol{I}}, r) = r \cdot \underbrace{\begin{cases} 1 & , \exists\{x, y\} \in \mathcal{N}_I \notin \bar{\Omega} \\ 0 & , \exists\{x, y\} \in \mathcal{N}_I \in \bar{\Omega} \end{cases}}_{\boldsymbol{in-polygon\ query}} \tag{5}$$

The proposed penalty term includes an *in-polygon* query routine in which each point of the population is checked to be either in or out of the domain $\bar{\Omega}$. The full reference for the algorithm can be verified in [7]. If any of the query points presents to be laying on $\partial\bar{\Omega}$ or outside, the objective function is penalized by the factor of r. The graphical representation of the objective function is illustrated in Figure 1.

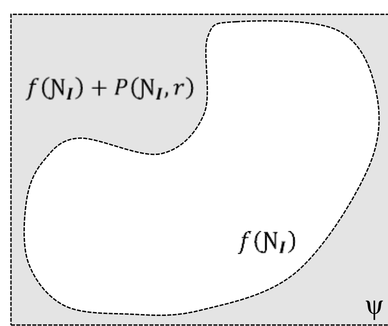

Figure 1. Graphical representation of the proposed penalized objective function

2.4 Optimization Approach

For optimizing the objective function, an all-node approach is considered where the positions of all inner nodes are moved simultaneously within each generation.

For each population of random interior nodes, a Delaunay Triangulation (DT) is performed to obtain nodal connectivity of the triangular partition $\mathcal{T}$. Due to the random nature of the population generation, it is not possible, with the proposed algorithm, to maintain the nodal connectivity throughout the optimization process, leading to discontinuities of the fitness landscape. Considering this, the use of a GA shows to be more promising than a gradient based optimization algorithm. By using a GA, it would then be possible to deal with the eventuality of a non-smooth fitness landscape as well as explore the extremities of the variable search domain, in which the mesh metrics can present abrupt changes. The pseudocode of the proposed algorithm is presented in Figure 2.

START Mesh Optimization Routine

1. *Define: (by user input **OR** feature extraction from base mesh to optimize)*
   a. *Mesh Boundary Nodes*
   b. *Number of interior Nodes*
2. *Define:*
   a. *Penalty parameter r*
   b. *Genetic Algorithm parameters*
3. *Optimization Process → Genetic Algorithm*
4. *Post-Processing and results evaluation.*

***END** return optimized mesh*

Figure 2. Pseudocode of the proposed optimization routine

Further detailing step (3), the GA carries out several operations to assess the mesh metric and return its value. These operations are presented in the following pseudocode. For further explanation on the definition of a generic GA, consult, [8].

START Mesh Metric Assessment

1. *Distribute* $\boldsymbol{n}$ *points randomly over the domain-inscribing searching space* $\psi \in \mathbb{R}^2$
2. *For the configuration* $\boldsymbol{P_n}$ *perform a Delaunay Triangulation to obtain partition* $\boldsymbol{\mathcal{T}}(\boldsymbol{P_n})$
3. *Calculate area* $\boldsymbol{A_i}$ *of each triangle* $\boldsymbol{\mathcal{T}_i} \in \boldsymbol{\mathcal{T}}(\boldsymbol{P_n})$ *and variance* $\sigma^2(\mathbf{A})$
4. *Perform in-polygon query*
5. *Calculate fitness value based on **Eq.3**, **Eq.4** and **Eq.5***

***END** return fitness value*

Figure 3. Pseudocode for metric assessment at each generation

3. NUMERICAL RESULTS

In this section, numerical results are described to illustrate the applicability of the proposed method, displaying the optimization of test meshes and its applicability to non-convex domains. To verify the smoothness of the fitness landscape for the considerd quality metric, a simple 1 free node mesh optimization was performed. Figure 4 shows the fitness landscape contour plot as well as the evolution of the initial distibution of individuals until convergence of the considered example mesh.

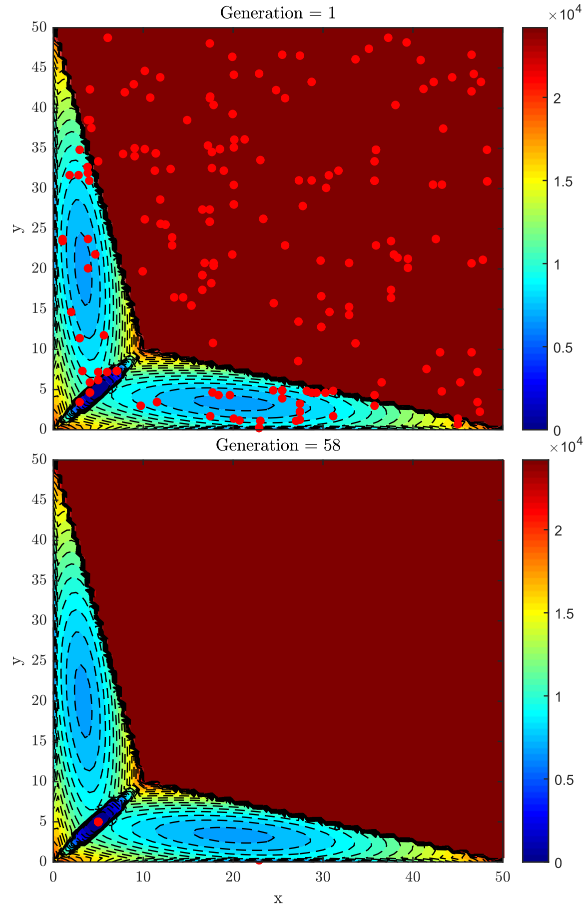

Figure 4. Contour plot of fitness value for a 1 free noded. Population distribution at generation 1 and after convergence

The proposed algorithm successfully converged to the global optimal node location leading to the mesh illustrated in Figure 5.

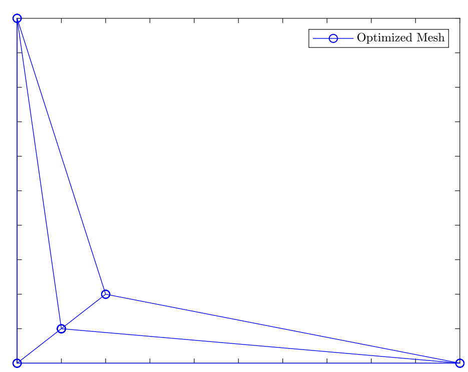

Figure 5. Resulting optimal mesh

The proposed algorithm was then tested to optimize meshes generated previously through a commercial mesher software and with higher number of internal nodes. Figure 6 illustrates the base and optimized mesh for the considered domain.

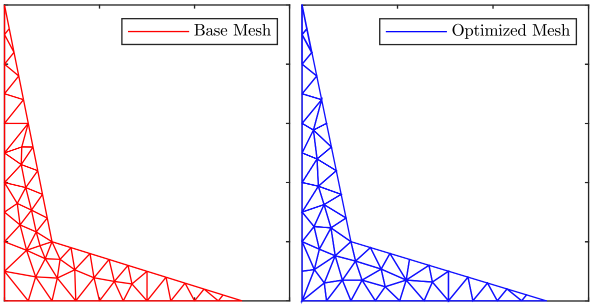

Figure 6. (left) Initial base mesh and (right) mesh after optimization

To quantitatively evaluate the effect of this adaptivity on the element areas, an *error map* is presented in Figure 7. This map shows the sample standard deviation of the each element of the mesh.

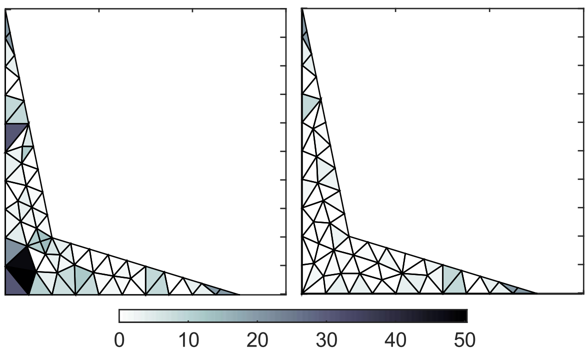

Figure 7. Area standard deviation of (left) Initial base mesh and (right) mesh after optimization

4. CONCLUSIONS

The aim of this study was to develop a mesh optimization method capable of handle simple non-convex polygons efficiently. A genetic algorithm was proposed to handle possible non-smoothness of the objective metric function. To deal with to impossibility to use constraint conditions to properly define the non-convex domain, a penalty based objective function with an *in-polygon query* was implemented.

The proposed method showed to be capable of optimizing a given mesh considering a mesh uniformity metric. Numerical results clearly show an increase in the number of elements with low area deviation, which suggest improvement of the mesh considering the mesh metric employed.

The implemented method can also be used either as an optimizer or as an optimal mesh generator. As mesh generator, the algorithm only requires information about boundary discretization and number of desired interior nodes. Other evaluation metrics can easily be incorporated into the developed algorithm.

AKNOWLEDGMENTS

The authors would like to acknowledge the Portuguese Foundation for Science and Technology for supporting this research through the PhD scholarship 2021.06449.BD; through IDMEC, under LAETA (Project No.UIDB/ 50022/2020); and through CDRSP, under ARISE (Project No.UIDB/ 04044/2020).

REFERENCES

[1] J. Kim, 'An efficient approach for solving mesh optimization problems using Newton's method', *Math Probl Eng*, vol. 2014, 2014, doi: 10.1155/2014/273732.

[2] J. Kim, M. Shin, and W. Kang, 'A derivative-free mesh optimization algorithm for mesh quality improvement and untangling', *Math Probl Eng*, vol. 2015, 2015, doi: 10.1155/2015/264741.

[3] C. S. Chong, H. P. Lee, and A. S. Kumar, 'Genetic algorithms in mesh optimization for visualization and finite element models', *Neural Comput Appl*, vol. 15, no. 3–4, pp. 366–372, Jun. 2006, doi: 10.1007/s00521-006-0041-2.

[4] G. Giorgi, A. Guerraggio, and J. Thierfelder, 'Chapter II - Convex Sets. Convex and Generalized Convex Functions', in *Mathematics of Optimization*, 2004, pp. 23–200.

[5] P. M. Knupp, 'Algebraic Mehs Quality Metrics', 2001. [Online]. Available: http://www.siam.org/journals/sisc/23-1/37149.html

[6] Y. Fujiwara and H. Sawai, 'Evolutionary computation applied to mesh optimization of a 3-D facial image', *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 113–123, 1999, doi: 10.1109/4235.771165.

[7] K. Hormann and A. Agathos, 'The point in polygon problem for arbitrary polygons', 2001. [Online]. Available: www.elsevier.com/locate/comgeo

[8] S. Katoch, S. S. Chauhan, and V. Kumar, 'A review on genetic algorithm: past, present, and future', *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.