Optimization methodology for the material assignation in bioprinted scaffolds to achieve the desired stiffness over time

Rubén Paz*
Universidad de Las Palmas de Gran Canaria, Departamento de Ingeniería Mecánica, Las Palmas de Gran Canaria, 35017, Spain, e-mail: ruben.paz@ulpgc.es

Mario D. Monzón
Universidad de Las Palmas de Gran Canaria, Departamento de Ingeniería Mecánica, Las Palmas de Gran Canaria, 35017, Spain, e-mail: mario.monzon@ulpgc.es

Abstract:
The optimum scaffold for tissue engineering must guarantee the mechanical integrity in the damaged zone and ensure an appropriate stiffness to regulate the cellular function. For this to happen, scaffolds must be designed to match the stiffness of the native tissue. Moreover, the degradation rate in the case of bioresorbable materials must be also considered to fit the tissue regeneration rate. This paper presents a methodology based on Design of Experiments, Finite Element Analysis, metamodels and Genetic Algorithms to optimize the assignation of material in different sections of the scaffold to obtain the desired stiffness over time and comply with the constraints needed. The method applies an initial sampling focused on a modified Latin Hypercube strategy to obtain data from the simulations. These data are used in the next stages to generate the metamodels by using Kriging. The predictions of the metamodels are used by the genetic algorithms to find the best estimated solutions. Different runs of the genetic algorithm drive the sampling, improving the accuracy of the surrogate models over the optimization process. Once the accuracy of the metamodels estimates is sufficient, a final genetic algorithm is applied to obtain the optimum design. This approach guarantees a low sampling effort and convergence to carry out the optimization process. The method allows the combination of discrete and continuous design variables in the optimization problem, and it can be applied both in solid and in hierarchical-based geometries.

Keywords:
Bioprinting, Material Assignation, Optimization, Genetic Algorithms, Metamodels, Finite Element Analysis.

1. INTRODUCTION

Additive Manufacturing (AM) is defined as a “process of joining materials to make parts from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing and formative manufacturing methodologies” (ISO/ASTM 52900:2015. Additive manufacturing -- General principles –Terminology). The evolution of these techniques in the last decades has led to the development of new technologies with capability to fabricate customized biodegradable structures suitable for regenerative medicine and tissue engineering [1]. In this field, the term “scaffold” is used to define a structure that serves as support for the growth of extracellular matrix of the damaged tissue.

When designing a scaffold, several requirements must be achieved to success [2,3]. First of all, the material must be biocompatible and promote cell function (attachment, migration, proliferation, etc.). Secondly, the scaffold must also degrade to allow the body cells the production of their own extracellular matrix, thus replacing the scaffold over time. Moreover, the resulting products of the degradation process must be non-toxic and easy-to-remove from the body. All these characteristics can be controlled through a suitable selection of materials. On the other hand, the scaffold must have a cellular structure with a large volume fraction of interconnected pores. The greater the surface of the scaffold, the better the cell adhesion is. However, for some tissues, larger pores can reduce the cell aggregations on the scaffold edges [4]. The pore shape and size are other important factors [4,5]. According to [6], pore size gradients improve the cell seeding efficiency and also achieve a better distribution of cells through the scaffold. Regarding the overall scaffold, the first design requirement is the mechanical integrity for handling it during surgery. Once the scaffold is placed in the body, its main goal is to maintain the microscale mechanical integrity, but its stiffness is also crucial to regulate the cellular function [7,8] such as cell differentiation [9] and cell migration. While scaffolds with an inappropriate stiffness may frustrate the regeneration process, scaffolds with tailored stiffness improve the efficacy of biochemical stimuli [10] since cells adjust their internal stiffness to match that of their substrate [11]. For this reason, the ideal scaffold must bear the corresponding load and also match the stiffness of the native tissue [10]. These tuned scaffolds are difficult to be manufactured with conventional fabrication techniques. However, AM can easily reproduce custom-tailored scaffolds [12], as different geometrical and architectural configurations can be easily applied [13] as well as materials.

Different optimization methods have been already applied to optimize porous structures for tissue engineering. Hollister *et al*. [14] proposed an optimization algorithm in which the algorithm optimizes the elastic modulus of the material of the scaffold and the geometry of the unit cell structure to match the native stiffness. This optimization is carried out by assessing two time points and assuming some simplifications: the initial time (in which the authors assumed that only the scaffold exists), and the long-term time (in which they assumed that only the regenerated tissue is present and it occupies the pores of the initial scaffold). On the other hand, the cell geometry consists in a solid cube that is intersected by cylindrical pores, creating an interconnected porous structure. The algorithm optimizes the diameter of the cylindrical pores, thus having a continuous design variable for each Cartesian axis. The method optimizes the diameters and the elastic modulus of the material so that the scaffold itself and the regenerated tissue match the desired stiffness while maintaining a specific level of porosity.

This method was later improved in [15]. The authors proposed a microstructural topology optimization method to obtain the cell unit with the best distribution of material to match the desired target stiffness of the cell unit. The method is based on the voxel discretization of the cell unit. For each voxel, a density value is applied with values from 0 to 1 (0 represents a void and 1 a solid voxel). The density of each element defines the contribution to the overall stiffness objective (in the case of values between 0 and 1, the stiffness is calculated using the solid isotropic microstructure with penalization material model [16]). The method optimizes the value of the density of each voxel, keeping the porosity within the desired limits. Moreover, the algorithm includes a refinement loop so that the finite element mesh (voxels discretization) is improved over the iterations. In such a way, the final output will have more resolution,

achieving the optimal geometry of the cell unit. Despite the clear advantages of combining topology optimization with porosity constrains to obtain the desired unit cell stiffness, this method has some limitations. Firstly, the optimization may lead to complex unit cells that can only be 3D printed with high resolution AM technologies such as material jetting or VAT photopolymerization. However, extrusion-based AM technologies cannot be applied due to a lower resolution. This is an important drawback since extrusion-based techniques are the most promising for regenerative medicine due to the higher range of biopolymers that can be processed [17,18]. Apart from this, topology optimization can lead to non-interconnected pores, which is a clear disadvantage for the cell growth. This is a drawback compared to [14] since the orthogonal cylindrical pores ensured the connectivity. On the other hand, the optimization algorithm depends on the initial properties of the selected material. This means that the method allows the cell unit optimization according to the material previously chosen, but it does not optimize the material selection process. On the contrary, in [14] the method allowed the optimization of the elastic modulus of the scaffold material, but in practice the list of available materials is limited and therefore the optimal elastic modulus will probably not be feasible.

Another relevant work related to scaffolds optimization is depicted in [19]. In this work, the authors used an evolutionary algorithm to optimize the material assignation of an extrusion-based scaffold. They assumed a desired stiffness of the overall scaffold (scaffold Young's modulus) at three different times (early, middle and late) and in two different locations along the scaffold. Genetic algorithms (GAs) were applied to optimize the material assignation of each strut within the scaffold to achieve that stiffness over time. Five different materials were taken into account and the elastic modulus of all of them over time was introduced as input data. As the total number of struts was 96, each different design proposed by the algorithm ('individual' for the genetic algorithm) was defined by 96 variables (each variable will represent the material of a single strut). Since the optimization was carried out by using a database of 5 materials, the values of the design variables could vary from 0 to 4 (integer numbers), each one representing a different material. The genetic algorithm was coupled with the ANSYS software to automatically accomplish the Finite Element Analysis (FEA) of each design proposed by the optimization method. The results of the FE analysis at each time point were used to assess the 'fitness function' of each individual. This fitness function is basically calculated as a constant value minus the difference between the desired stiffness and the obtained one for the three degradation times and at the two locations along the scaffold. Therefore, the objective is to maximize this fitness function so that the scaffold achieves the desired stiffness.

The genetic algorithm generates an initial random population of five individuals. This low value was selected to be able to accomplish the optimization in a standard workstation. Afterwards, these five designs are evaluated by FEA and the results are used to calculate the fitness function of each one. Depending on the value, a probability of selection is assigned to each individual, so that the fitter designs (compared to the desired stiffness at the different degradation times) will have a higher probability. In the next step, the five selected individuals are subjected to crossover. This consists in selecting two individuals and, by combining them, generate two new 'offspring' individuals to replace the initial ones. Finally, mutation is randomly applied, thus changing one of the design variables of some of the individuals of the new population. These steps were repeated in a loop to emulate the natural selection process in the

nature: the better adapted to an environment will survive and reproduce, so that over many generations, the process of natural selection leads to evolution (which, in this case, means optimization).

This methodology combines FEA with a powerful searching tool such as genetic algorithms, allowing the selection of the 'best' combination of materials to achieve the desired stiffness. As depicted in the literature [20], the integration of material selection within FEA predictions or optimization tools is, without doubts, an important step forward in the field of Computer-Aided Tissue Engineering. However, the use of this type of evolutionary algorithms requires high computational times. For example, in the case study presented in [19], each iteration of the genetic algorithm required the simulation of five different designs at three time points, meaning 1500 FE analyses to carry out the optimization. This number of simulations may be unfeasible, especially for complex problems in which each simulation may take several hours.

A new methodology is presented in the following sections to improve the limitations observed in the literature review.

2. METHODS

2.1. General approach

The methodology presented in this work is based on a previous approach [21] and inspired by the work of Heljak *et al.* [19], reviewed in the previous section. The main novelty introduced is the use of metamodels to reduce the computational time needed for the evaluation of the fitness function of the individuals generated during the genetic algorithm evolution. Figure 1 summarizes this idea.

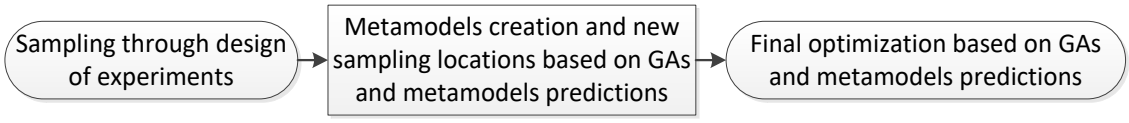

Figure 1. General concept of optimization based on sampling (FEA), metamodels and genetic algorithms.

To accomplish this, a first stage of design of experiments (DOEs) is applied to obtain information about the problem. In other words, different designs are evaluated by FEA and the results are used to create a database of constraints and objectives according to the optimization problem (Figure 2). The number of designs evaluated in this stage will be in most of cases '3+n' (being 'n' the number of design variables of the optimization problem). The DOE applied consists in first evaluating the designs with all the variables with the minimum value, all the variables with middle value and all the variables with the maximum value. In case that all the design variables are discrete and with only two possible values, the middle point is omitted. Afterwards, a modified Latin Hypercube is carried out by adding 'n' new designs. The location of these points is initially determined by a standard Latin Hypercube ('lhsdesign' function in MatLab) that divides each dimension of the search domain in 'n' equal regions and randomly determines the location of the first point. The second point is added by avoiding the 'columns' and 'rows' of the previous point in the 'n^n' grid ('n' dimensional space) and maximizing the minimum distance between points. This is repeated until having added the 'n' sampling

points. As the 'lhsdesign' function works with values between 0 and 1, the sampling values previously obtained are adapted to the real search domain by using the limit values of each variable. Afterwards, the algorithm checks whether the point is in a boundary region of the 'n^n' grid and in that case, the sampling is moved to the bounds to increase the space coverage of the sampling points, thus reducing the extrapolation and promoting the interpolation (more accurate) in subsequent stages. This modification to the standard Latin Hypercube is also applied for discrete variables with the peculiarity that the final value is round to obtain an integer value.

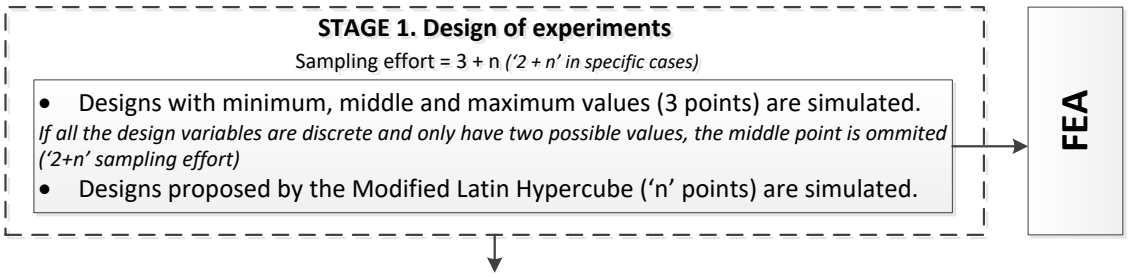

Figure 2. Flow chart of stage 1 of the optimization methodology.

With this database, the second stage of the methodology (Figure 3) applies GAs to find the best design according to the desired objective. However, before the running of the GA, a Kriging metamodel is generated (using the data previously obtained) for each constraint and objective of interest (e.g. stiffness, safety factor, etc.). Once the metamodels are created from the simulated database, they can be used to predict the results of any design without performing the FEA. This capability is used within the GAs to evaluate the fitness function of each individual without carrying out FEA simulations. Therefore, the computational time can be significantly reduced compared to the reference work [19]. Moreover, the parameters of the GA such as the population size can be easily modified to guarantee the convergence and without significant changes in the computational time since the fitness function evaluation is calculated from the estimates of the metamodels, not from the FEA results. Using this concept, the methodology takes advantage of a GA to find the best design according to the metamodel estimations. The best design is then simulated by FEA and the results are added to the database so that in the next iteration, the metamodels will be updated with the new data, thus improving their accuracy over the optimization process. Note that in the first 'n' iterations of the second stage, a proximity penalty is applied in the fitness function evaluation when the individuals are too close to the already existing data. The aim of this penalty is to avoid the evolution of the GA towards locations that has been already explored, thus encouraging a more uniformly distributed sampling across the search domain. More details will be explained in section 2.4. The predicted and simulated values are also compared in the final iterations of the second stage to assess the accuracy of the metamodel. Therefore, the estimation error is used as a control parameter to proceed or not to the final stage of the methodology. If the main absolute percentage error (MAPE) of the predicted objectives and constraints (compared to the FEA results) is higher than 5%, another iteration is applied.

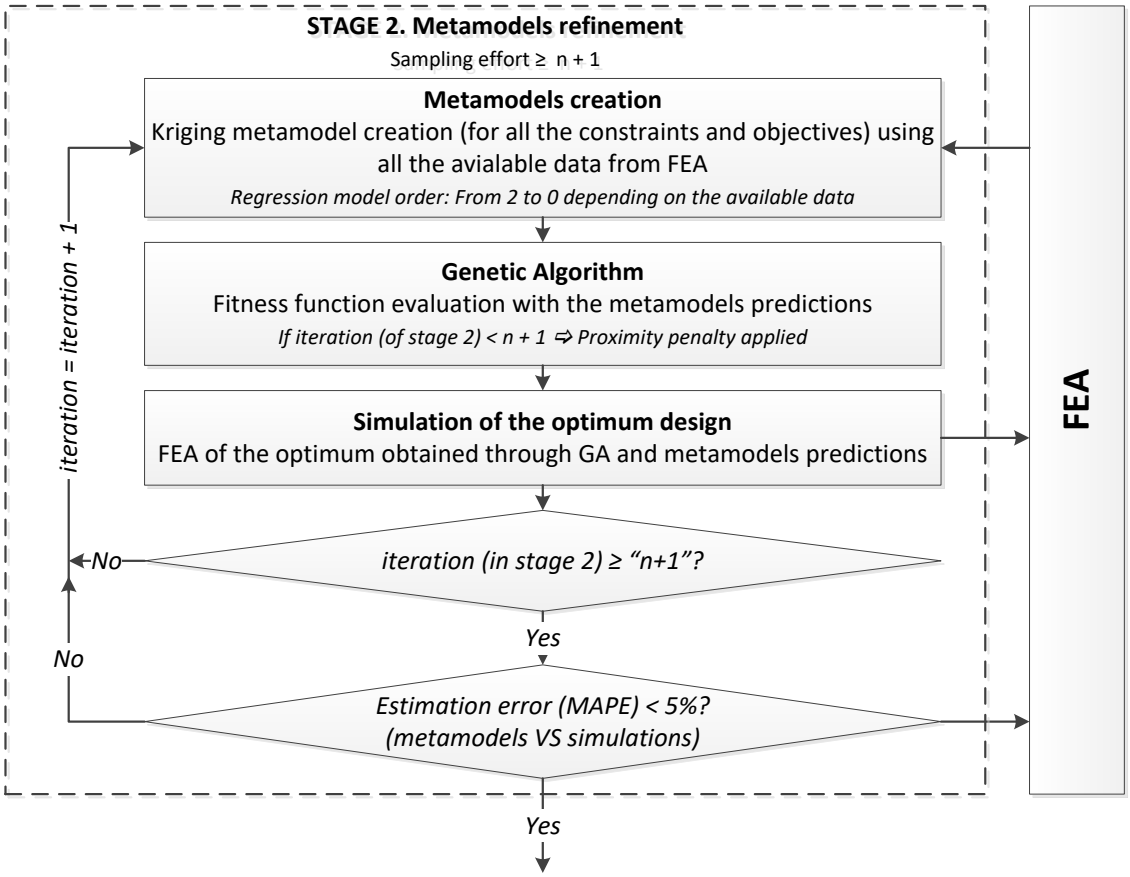

Figure 3. Flow chart of stage 2 of the optimization methodology.

Once the MAPE of the metamodels predictions is lower than 5%, the final phase of the algorithm is accomplished (Figure 4). This consists in applying a GA combined with the updated metamodels to predict the fitness function values. The optimum predicted by the GA and metamodels is simulated by FEA and if it improves the best design achieved in previous stages, it will be considered the optimum. Otherwise, the algorithm updates the metamodels with the last design simulated and repeats the process in a loop until the optimum is achieved. Moreover, if this final stage applies more than '5+n' attempts without success in the improvement of the previous optimum, then this previous optimum will be considered the final optimum design.

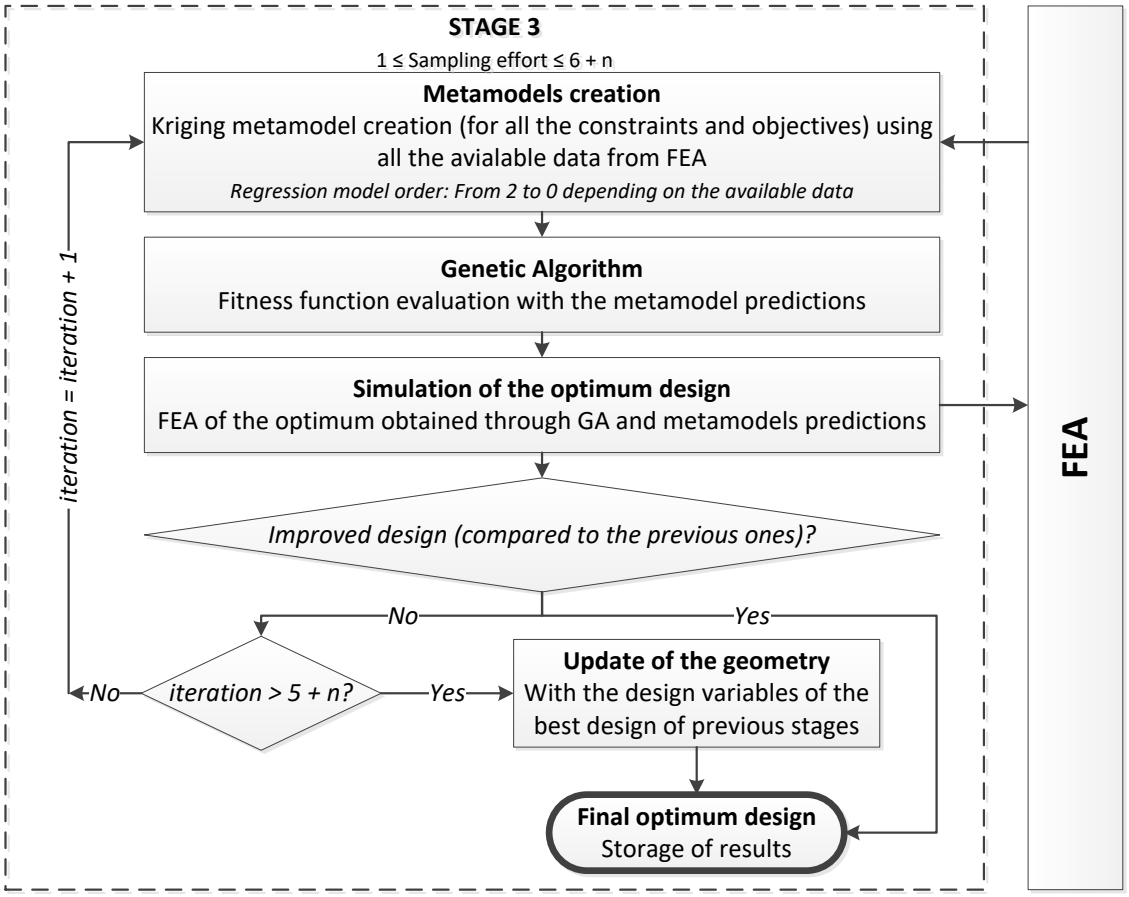

Figure 4. Flow chart of stage 3 of the optimization methodology.

Although this methodology is based on the proposal depicted in [21], some significant modifications were implemented to be able to deal with different time points, material assignation, user-defined objective functions and other requirements needed for this specific application. For example, in [21] the GAs were coded for continuous variables, whereas in this case the variables can be continuous, discrete or a combination of both. Moreover, the design variables reserved for material assignation are not associated with design variables of the CAD model, but material properties that must be applied to the solids in the FEA. Therefore, the internal workflow to manage these variables is different. The following sections explain the main implementations and parameters used in the methodology and GAs to provide the required features for this specific application.

2.2.Data input

The methodology has been implemented within SolidWorks 2016 software by using the Application Programming Interface. After running the code, the main window asks for the output directory and file name, the number of design variables (‘n’, which will be used to determine the sampling effort in the different stages of the algorithm), the number of discrete design variables, the number of discrete design variables to assign materials, the number of materials, the number of time points for the degradation process, the number of constraints for the optimization definition, the number of studies (to apply different analysis if needed), the number of configurations (to assess different

shapes of a unique design, mainly useful for 4D printing optimization [21]) and the number of desired runs to repeat the same optimization process many times (useful to compare the results and analyze the behavior of the optimization methodology). According to the code structure, the continuous design variables must be the first to be defined, while the discrete variables for material assignation must be the last ones. Also, note that the discrete variables can include discrete variables associated with the CAD design (e.g. number of instances in a matrix operation), or discrete variables for material assignation. In that case, the variables reserved for material assignation must be always defined at the end. On the other hand, the code is prepared to deal with discrete variables defined as integer numbers with 1 unit step between values.

Afterwards, the material properties (elastic modulus, Poisson's ratio, yield strength and density for each material and for each time point), upper and lower limits of each design variable, limits and feasible zone of the optimization constraints, mesh element size for each analysis, number of objective sensors, definitions of the objective function and associated analysis of each objective sensor (each sensor will be associated to a unique FE analysis) are required.

2.3.Flow chart of the FEA

The FEA of each different design requires several steps that are summarized in Figure 5. First of all, the global variables related to the CAD model are updated in the equation manager in SolidWorks. Afterwards, the geometry is updated if any of these variables has been changed. Subsequently, several steps are applied for each analysis. Note that the methodology is prepared to work with different analysis that can provide information needed for the optimization problem (constraints or objectives). The first step for each analysis is to create the mesh with the element size dimension introduced during the input data request. In the case that the geometry has not been modified, the mesh generation is not needed as it was already created in the previous design. Subsequently, for each degradation time, the following steps are carried out:

- In the case of having design variables reserved for material assignation, the methodology allocates the material properties to each solid. For example, if design variable 1 (reserved for the material assignation of solid 1) is '4', then the mechanical properties of material 4 (at the corresponding time point), introduced in the data input, are assigned to this solid.
- The methodology runs the analysis.
- Once the FEA analysis is completed, the results obtained in the constraints and objectives sensors are stored.

Once these steps are carried out for all the degradation times, the methodology assesses the fitness function of the design evaluated by using the objective function introduced in the data input. Moreover, a penalty is applied for each unfulfilled constraint. Section 2.5.2 will give more details about this.

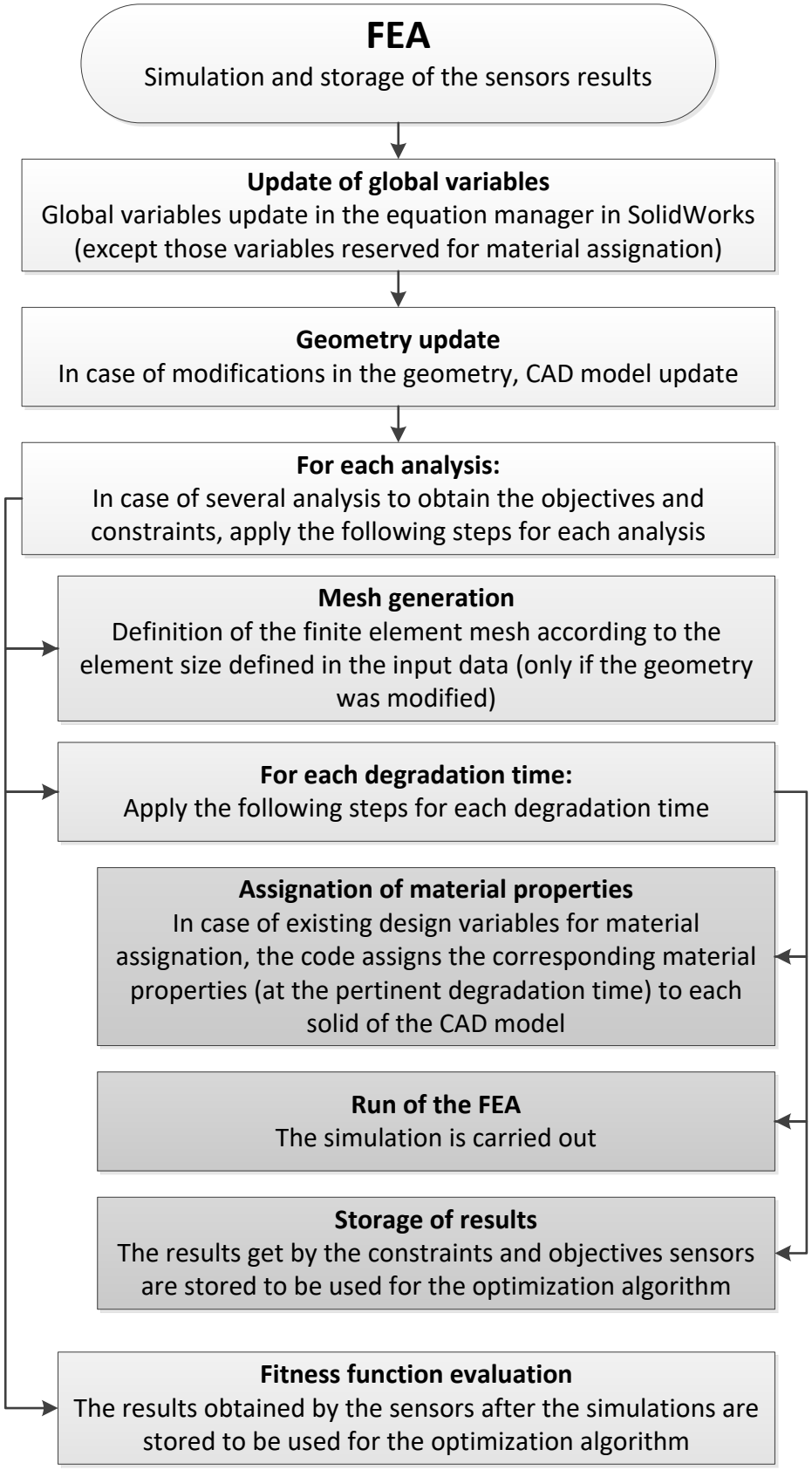

Figure 5. Flow chart of the steps carried out during the FEA of each design.

2.4.Kriging metamodels

The Kriging method is used to predict the results within the GAs from the available data of the FEA. This interpolation/extrapolation method was implemented in the methodology through the MatLab Kriging Toolbox developed by Lophaven *et al.* [22]. The generalized exponential correlation model was selected since it is the most commonly used when the spatial correlation between data is unknown. Regarding the regression model, this toolbox allows the usage of a polynomial function from 2 to 0-order. Taking into account that the highest the order, the more accurate the predictions are, the methodology attempts to first create the 2-order metamodel and if it fails, it will progressively reduce the order until the metamodel is created. With this strategy, the highest possible order of regression model (according to the available data) is always applied.

2.5.Genetic algorithms

The stages of the genetic algorithms applied in this work are summarized in Figure 6. The main parameters used are explained in the following sections.

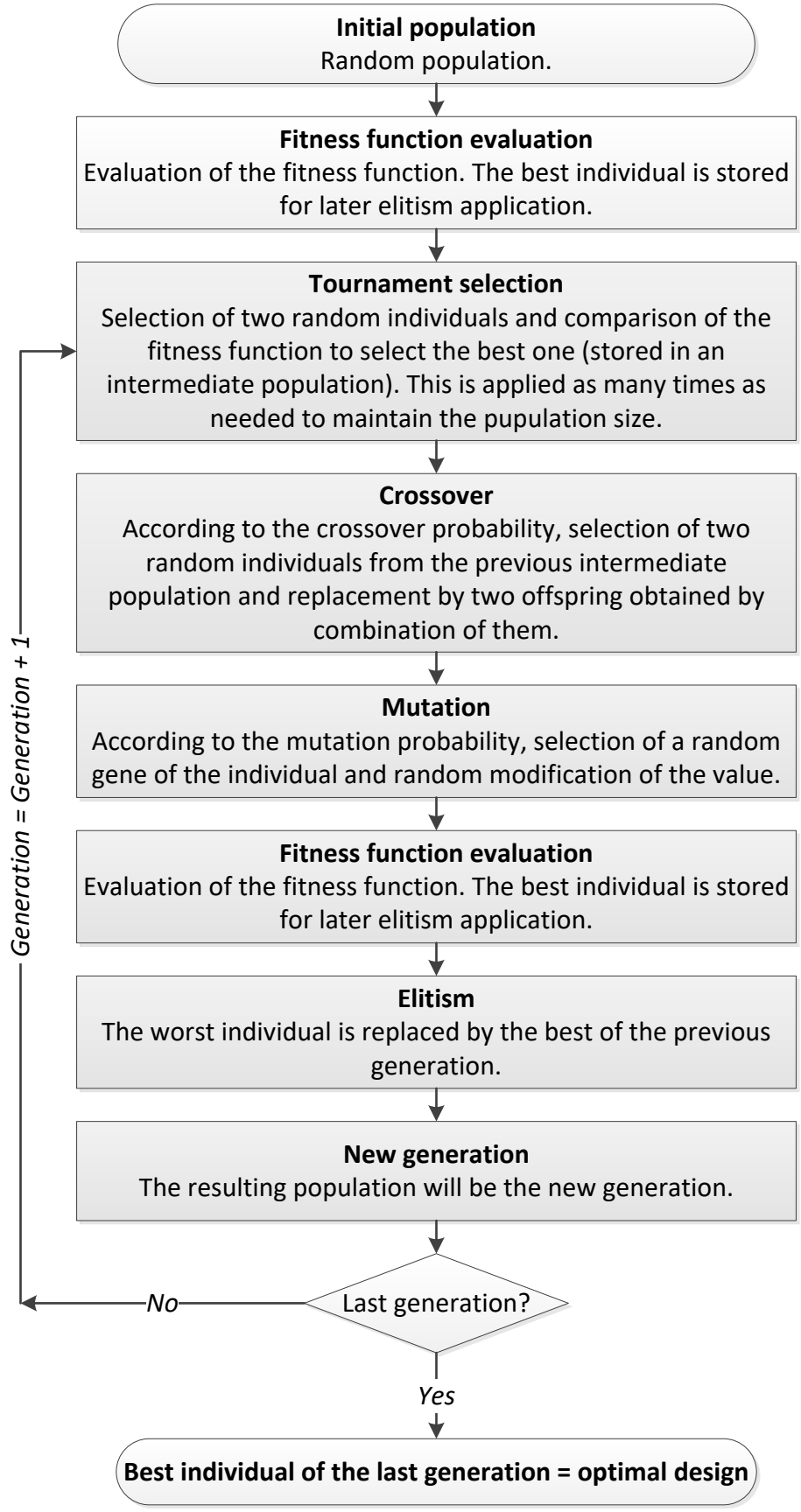

Figure 6. Flow chart of the genetic algorithms used in this work.

2.5.1. Population size and number of generations

The population size was fixed in 100 individuals and the number of generations in 100. Therefore, each GA evolution requires the evaluation of a total of 10000 individuals. Thanks to the use of metamodels, the evolution of each GA with 10000 individuals requires just a few seconds in a workstation with an Intel® Core™ i-5-6300HQ Processor @ 2.3GHz.

2.5.2. Initial fitness function evaluation

The fitness function is obtained from the definition of the objective function and the addition of some penalty terms.

The objective function can be user-defined according to the optimization problem, but it is necessary to follow a specific nomenclature. The pre-processing of the FEA model requires the definition of sensors to read the desired values from the simulation results (e.g. maximum displacement). Depending on the nature of the sensor for the optimization problem, it will be defined as 'objective' or 'constraint'. Several objective and constraint sensors can be declared, each one with its corresponding identifier (integer values starting from 1). Therefore, in the objective function definition, the values of the objectives sensors will be declared as 'obj(j,m)', being 'j' the identifier of the objective sensor and 'm' the time point (both integer values). Section 3 will show more details about the objective function definition through a case study.

Apart from the free definition of the objective function in the corresponding prompt window, the user must also specify if the objective must be maximized or minimized by selecting the appropriate option button. Some penalty terms (positive for minimization problems and negative for maximization) are automatically added to the objective value to assess the fitness function. This happens when the designs do not comply with the optimization constraints.

On the other hand, note that the fitness function within the GAs is evaluated by using the estimated results of the objectives and constraints (through the metamodels). Moreover, in the second stage of the proposed methodology, a proximity penalty is applied in the first 'n' iterations (Figure 3). The aim of this penalty is to worsen the fitness function value of those individuals close to points already simulated. Consequently, the GAs will converge towards solutions away from the existing sampling points. When this concept is applied in several GA runs, the result is a well-distributed sampling across the search domain and in locations of interest according to the metamodels predictions. Therefore, this concept allows an efficient selection of the sampling points to improve the accuracy of the metamodels in areas with potential to be the location of the optimum design. To apply this, first a radius of influence is defined. Afterwards, the distance between each individual of the GA and the already existing sampling points is calculated. If the distance between the individual and any of the sampling points is lower than the radius of influence, the proximity penalty will be applied to penalize the fitness function of the individual. The mathematic details of this concept are depicted in [21] and the idea was based in on the resource sharing method used in multimodal optimization [23], where the fitness function value of each individual is reduced according to the number of individuals sharing the same search space (resource sharing).

The fitness function is also evaluated for each design simulated by FEA to show, at the end of the process, the quality of the designs that were actually evaluated by FEA.

2.5.3. Tournament selection

The tournament selection consists in creating an intermediate population by selecting two random individuals of the population and comparing the fitness function value. The best one is stored in the intermediate population. This is repeated 100 times to obtain 100 individuals in the intermediate population.

2.5.4. Crossover

The crossover is applied with 50% of probability. Half of the intermediate population is randomly preselected and combined to obtain offspring, while the other half is maintained. To carry out the crossover, two random individuals belonging to the previous group are selected. Afterwards, a random value 'α' (between -0.5 and 1.5) is generated and the two offspring are obtained as an arithmetic average of the two parents, using 'α' and '1- α ' as the weight of each parent:

$$Offspring_1 = \alpha \cdot Parent_1 + (1 - \alpha) \cdot Parent_2$$
$$Offspring_2 = \alpha \cdot Parent_2 + (1 - \alpha) \cdot Parent_1$$

These previous equations are used for each continuous variable. For the discrete variables, the crossover operator is different. A random value is generated (0~1) and if the value is lower than or equal to 0.5, '$Offspring_1$' will take the value of '$Parent_1$' and '$Offspring_2$' will take the value of '$Parent_2$'. However, if the random value is higher than 0.5, '$Offspring_1$' will take the value of '$Parent_2$' and '$Offspring_2$' will take the value of '$Parent_1$'.

2.5.5. Mutation

The mutation probability was fixed in 60%. For each individual of the resulting population, a random value is generated (0~1). If this value is lower than or equal to 0.6, the individual will mutate. The mutation consists in selecting a random gene of the chromosome (a random design variable) and slightly modifying the current value by adding the result of a random value between -0.5 and 0.5 multiplied by the maximum interval of that design variable:

$$Mutated\ variable = current\ value + (Upper\ limit - Lower\ limit) \cdot Random$$
$$Random = (-0.5 \sim 0.5)$$

In the case of discrete variables, the concept is the same but a round operation is applied to obtain an integer value.

Finally, a reparation loop is applied in the individuals that have been mutated. This consists in assessing if any of the design variables has exceeded the initial bounds. If this happens, the algorithm modifies the value to the nearest limit.

2.5.6. Fitness function evaluation

The fitness function is evaluated again following the same procedure explained in section 2.4.2.

The fitness function is also evaluated for each design simulated by FEA to show, at the end of the process, the quality of the designs that were actually evaluated by FEA.

2.5.7. Ranking and elitism application

In the last step of the GA, the fitness function of the resulting population is evaluated (through the Kriging metamodels) and sorted to replace the worst individual by the best of the previous generation (elitism).

Sections 2.4.3 to 2.4.6 are repeated in a loop until generation 100 is reached. The best design of the last population will be the optimum that will be simulated.

2.6.Workflow and software used

The methodology was developed in Visual Basic for Applications, which is one of the codes available for SolidWorks 2016 Application Programming Interface. Before running the code, the user must define the geometry, create the appropriate parameterization (link between design dimensions of the model and design variables represented by global variables in the equation manager) and define the sensors and analyses that will be considered in the optimization process. Once this preprocessing is ready, the optimization can be run. The code automatically opens a MatLab Windows Application to communicate during the optimization, since several tasks such as array sorting or metamodels generation/evaluation are carried out in MatLab.

3. CASE STUDY

A case study was carried out to test the methodology. The case study consisted in a 10mm diameter by 10mm height scaffold, under 1000N compression load. To facilitate the simulation and accuracy on the definition of the boundary conditions, double symmetry was applied. Therefore, the load applied in the model was 250N and the displacements normal to the planes of symmetry were constrained in the symmetry faces. Additionally, the displacements of the bottom face were also constrained in the vertical direction. The model was divided into 12 different sections of 0.8mm thickness, except the top one, with 1.2mm thickness (multiples values of 0.4mm, which is the layer thickness for the 3D printing). For each section, a design variable was applied for material assignation (Figure 7), leading to a total of 12 design variables (all of them discrete variables). On the other hand, 3 time points and a database with 5 different materials were considered for this case study. The elastic modulus and yield strength of the 5 materials (at the different time points) are depicted in Table 1. The Poisson's ratio was fixed in 0.3 for all the materials and time points. Note that the model is represented as a solid material, which means that the mechanical properties introduced would be associated with to the bulk properties of porous scaffolds. Therefore, the materials database may include examples with the same raw material but with different

configurations (infill strategies, layer height, etc.), thus leading to different mechanical properties.

The curvature based mesher was used with a maximum and minimum element size of 0.4 and 0.08mm (respectively) and a 1.5 element size growth ratio. Two-order tetrahedral elements were used. The resulting mesh had 34092 elements and 48596 nodes, with a maximum aspect ratio of 3.7.

The objective of the optimization is to achieve the stiffness over time depicted in Table 2 both for points 1 and 2 (see Figure 7). These values of stiffness of the overall scaffold were translated into displacements of the model. They are also depicted in Table 2. The restriction of the optimization problem is to avoid plastic deformation of the materials at any time point. Therefore, the safety factor must be higher than 1 in the entire model. Two objective sensors were defined in the analysis (Objective 1 and Objective 2, respectively) to read the displacements in points 1 and 2. Another sensor was also added to get the maximum safety factor (Yield stress / Maximum Von Mises stress) of the model (Restriction 1).

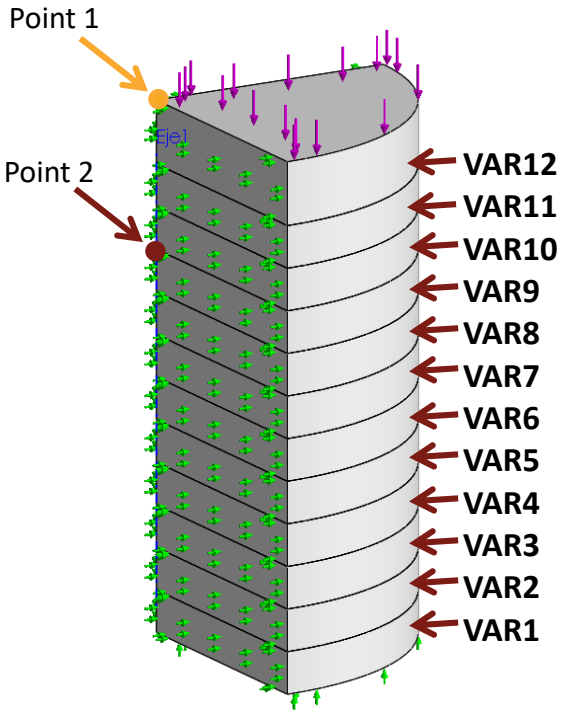

Figure 7. Model used for the optimization.

Table 1. Database of material properties at the different time points evaluated.

	Material 1		**Material 2**		**Material 3**		**Material 4**		**Material 5**	
	E (MPa)	Sy (MPa)	E (MPa)	Sy (MPa)	E (MPa)	Sy (MPa)	E (MPa)	Sy (MPa)	E (MPa)	Sy (MPa)
Time 1	100	20	200	20	300	40	400	40	800	90
Time 2	80	18	160	16	250	35	380	35	700	70
Time 3	50	16	100	14	200	30	360	32	400	60

Table 2. Desired stiffness over time in points 1 and 2 of the scaffold.

	Desired stiffness at Point 1 (N/mm)	**Objective 1 (mm)**	**Desired stiffness at Point 2 (N/mm)**	**Objective 2 (mm)**
Time 1	1818.18	0.55	2222.22	0.45
Time 2	1538.46	0.65	1818.18	0.55
Time 3	1111.11	0.9	1333.33	0.75

To minimize the difference between the desired displacements over time and the obtained ones, the objective function is defined as the quadratic error between them, so that the objective is to minimize this quadratic error. Therefore, the optimization problem can be defined as follows:

$$\boldsymbol{Min}\ F = (obj(1,1) - 0.55)^2 + (obj(1,2) - 0.65)^2 + (obj(1,3) - 0.9)^2 + \\ +(obj(2,1) - 0.45)^2 + (obj(2,2) - 0.55)^2 + (obj(2,3) - 0.75)^2 + Penalties$$

$$\boldsymbol{Constrained\ to}: \qquad Restriction\ 1\ (Safety\ Factor) > 1\ at\ any\ time$$

Where "obj(j,m)" represents the value of sensor "Objective j" (displacement in point 'j') at time point "m". Therefore, the objective is to minimize the difference between the achieved and the desired displacement for points 1 and 2 and at time points 1-3. By minimizing the previous expression, the optimum design will achieve the desired stiffness (approximately) over time in points 1 and 2. Note that the fitness function has a penalty term that is applied when the safety factor is lower or equal to 1.

Since the methodology is based on a stochastic method that can lead to different solutions, the optimization was carried out 30 times to assess the convergence of the methodology.

4. RESULTS

The first optimization run was completed in 26min after the evaluation of 34 different designs. The optimum design was the last one (design 34), having the combination of materials depicted in Figure 8 and a fitness function value of 0.0043. Table 3 shows the results of the restriction and objectives sensors of the optimum and Figure 9 the comparison between the desired and the obtained values of the objectives.

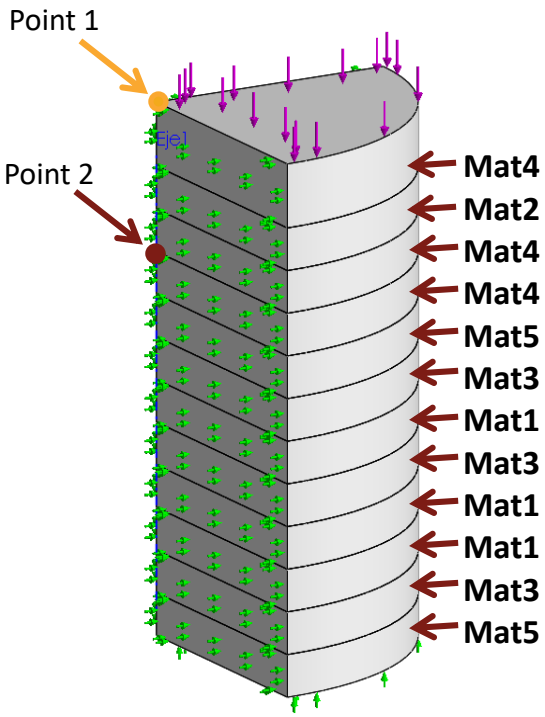

Figure 8. Optimum design of run 1.

Table 3. Objectives and restriction of the optimum design of run 1.

	Objective 1 (mm)	**Objective 2 (mm)**	**Restriction 1**
Time 1	0.5354 (desired=0.55)	0.4274 (desired=0.45)	1.4332 (>1)
Time 2	0.6424 (desired=0.65)	0.5209 (desired=0.55)	1.2702 (>1)
Time 3	0.9393 (desired=0.9)	0.7841 (desired=0.75)	1.1806 (>1)

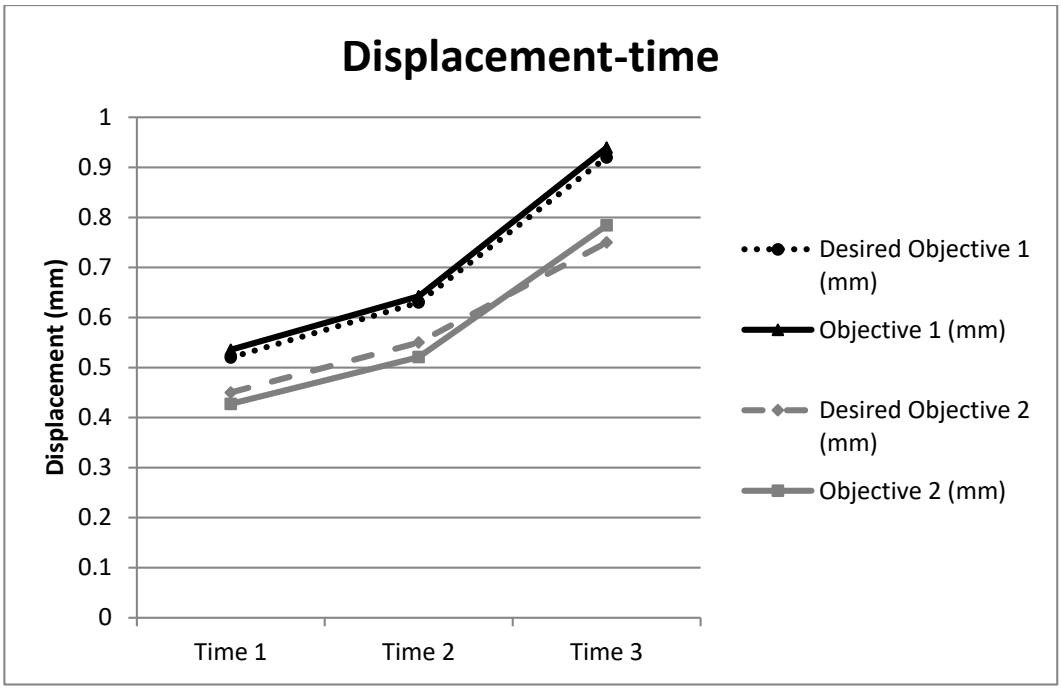

Figure 9. Comparison between the objectives obtained by the optimum of run 1 and the desired objective values.

Figure 10 shows the value of the fitness function of the designs evaluated during the optimization. Designs 15, 17 and 30 had higher values because the restriction was not complied due to a safety factor lower than 1 at time 3. This graph was zoomed in to visualize the smallest values of the fitness function (Figure 11). The first 15 designs were evaluated during stage 1 of the algorithm, designs 16-28 during stage 2 and designs 29-34 in stage 3. In this case, 6 attempts were needed in stage 3 to improve the best design of the previous phases (in this case obtained in stage 1).

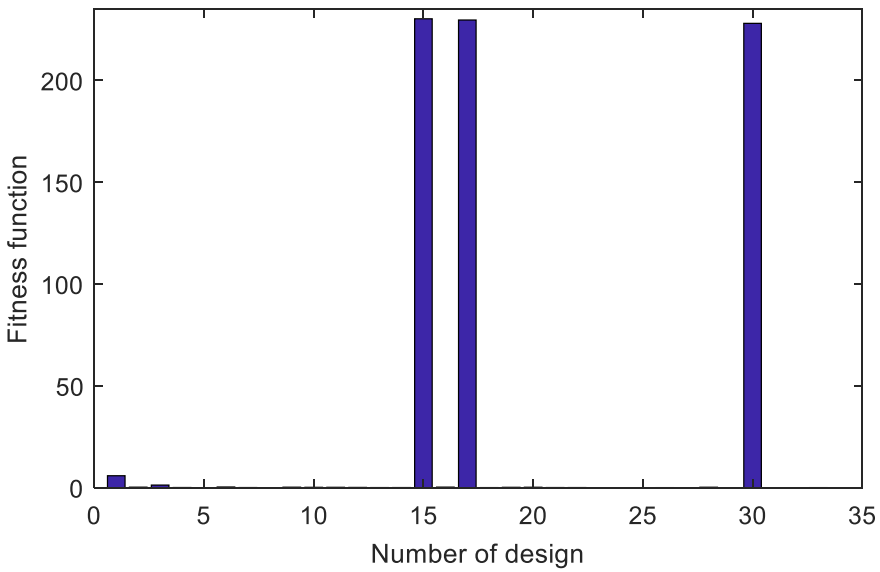

Figure 10. Fitness function of the designs evaluated during run 1.

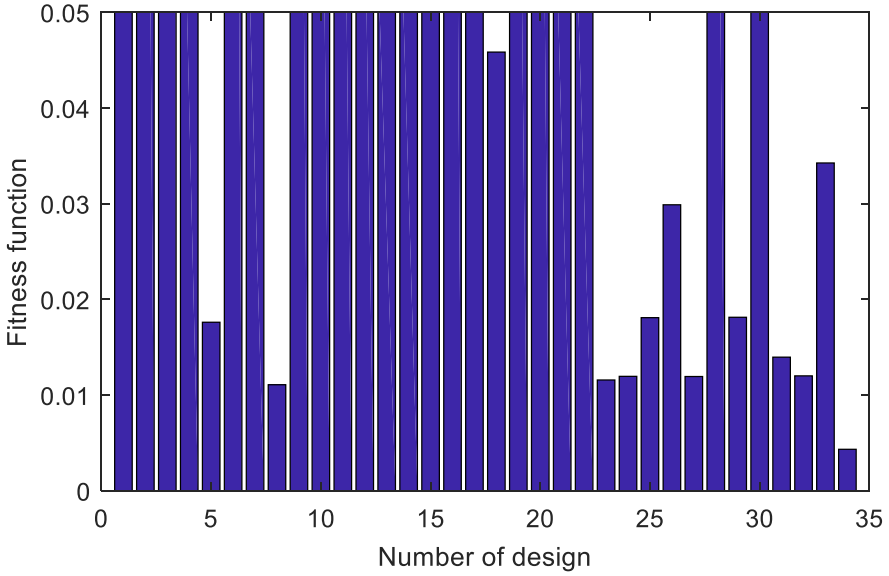

Figure 11. Detail of the fitness functions of run 1 (between 0 and 0.05).

The fitness function of the optimum and the optimization time of the 30 runs are shown in Figure 12.

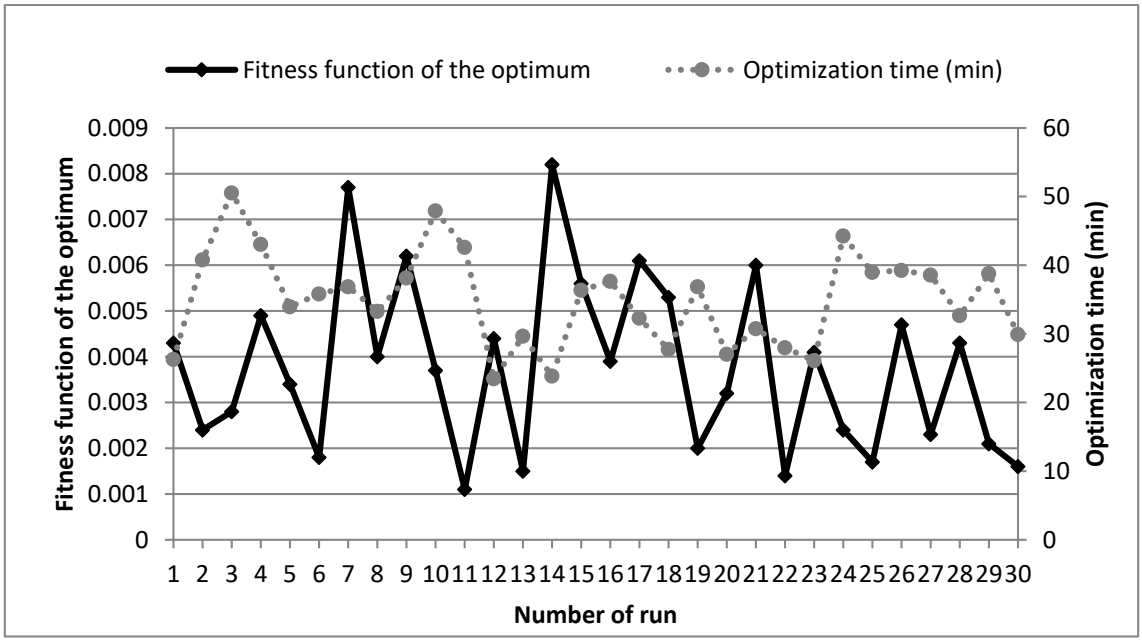

Figure 12. Fitness functions of the optimum and optimization time of the 30 runs.

5. DISCUSSION

The results of the 30 runs were statistically treated to assess the quality of the optimum. First of all, a box-plot was determined for the fitness function values of the 30 optimums, the safety factor (restriction) of the 30 optimums, the number of designs evaluated during the 30 runs, and the optimization time of each run (Figure 13). For the safety factor (SF), the average value of the 3 times was calculated for each optimum. It can be observed that there are no outliers, which means that the methodology converges

always to similar results. To statistically confirm this, first a Jarque-Bera test with 1% significance level was performed for each variable group of Figure 13 to assess the normality of the data. The Jarque-Bera tests concluded that the data came from a normal distribution (with an unknown mean and variance). Once the normality was confirmed, the average and sample standard deviation (SD) of the 30 samples were calculated (Table 4). Subsequently, the mean values were used to perform a one sample t-Test for each parameter (fitness function, SF, number of designs evaluated and optimization time) with a 1% significance level. For all the cases, the t-tests did not reject the null hypothesis that the data comes from a normal distribution with mean equal to the value previously calculated and unknown variance. Therefore, it was statistically proved that the methodology converges to similar results despite its stochastic nature.

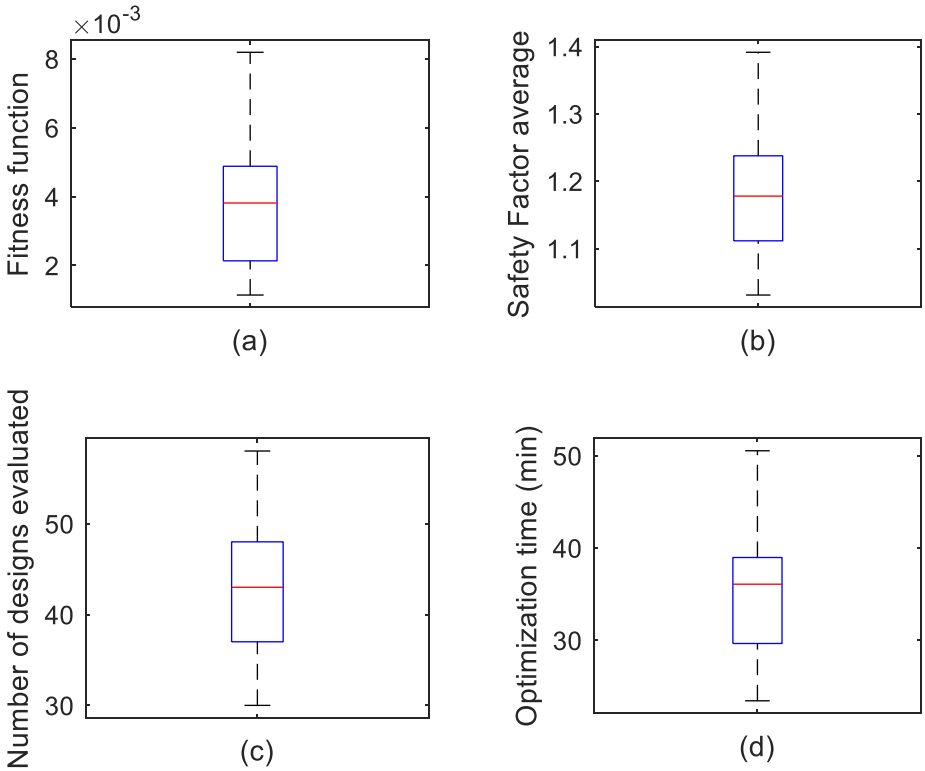

Figure 13. Box-plots of the fitness function, safety factor (average at 3 time points), number of designs evaluated and optimization time.

Table 4. Mean values of the fitness function and SF (average at 3 time points) of the 30 optimums, and number of designs evaluated and optimization time of the 30 runs.

	Fitness function	SF average (at 3 times)	No. designs evaluated	Optimization time (min)
Mean	0.0038	1.18	43.27	35.04
Sample SD	0.0019	0.09	7.37	6.96

Although this process enables the optimization of the material assignation taking into account the biomaterial degradation, it is needed an important database of material properties over time. To obtain this material database, mechanical tests at different degradation times should be carried out for different materials and scaffold configurations (infill strategies, layer height, etc.). On the other hand, different

simulation approaches or models could be also useful to estimate the mechanical properties over time and use the bulk properties as an input for this methodology. For example, in [24] the authors used a degradation model based on the loss of the molecular weight caused by hydrolysis, thus leading to reduced mechanical properties.

In this paper, a simplified solid 3D model has been used by applying the appropriate bulk properties in the different solid sections. Another option would be to model the hierarchical structure of the deposited filaments so that the bulk properties were not needed and therefore the amount of experimental tests could be drastically reduced. In this sense, the simplest approach could consist in drawing cylinders without considering the real path of the extrusion head, but with the appropriate porous density, as it was done in [19]. In this reference, the material of every single strut was optimized. However, this concept may not be feasible in practice as the material would have to be changed several times within every layer during the 3D printing. A more realistic approach could be to assign different materials to different layers, which is easier to implement in practice. In any case, the optimization methodology presented in this work could handle this approach since the concept is the same as the one applied in the case study. Some authors have proposed other methods [25,26] based on progressive element activation to generate the 3D printed geometry. These methods consist in dividing the geometry into voxels/elements and carrying out an activation of dormant voxels depending on the nozzle movements, so that only the voxels close to the nozzle during the printing are activated. This concept allows a more precise definition of the deposited geometry, but the generation of the deposited geometry can be very demanding in terms of CPU time, as well as the meshing and solving of the finite element analysis. In any case, the presented optimization methodology could be also applied to this type of geometries, especially if there are no variable associated with the design. In that case, these modelling techniques would have to be implemented in the proposed optimization methodology to tackle the optimization problem.

6. CONCLUSIONS

A new optimization methodology based on DOE, FEA, metamodels and GAs has been presented for the material assignation in scaffolds. This methodology allows the determination of the best combination of materials (from a database) to achieve the desired mechanical properties (such as stiffness) over degradation time. The methodology applies an initial DOEs to obtain data from the simulations and these data are used in the next stages to generate the Kriging metamodels. These metamodels are used to evaluate the fitness function of the designs proposed by the GAs. With this approach, the optimum search within the GAs is carried out without performing FEA, thus reducing the CPU time required for the optimum search. In addition, the second stage of the optimization algorithm includes some techniques to explore new areas of the search domain and to improve the metamodels accuracy. In fact, the final stage is only reached once a certain level of accuracy is achieved.

As demonstrated in the case study, this approach guarantees a low sampling effort to find the optimum design, which is especially interesting in the case of complex designs with complex FEA associated. Regarding the convergence of the methodology, the statistical analysis carried out revealed that the optimums were not different between them. Therefore, the methodology leads to similar designs despite the stochastic nature

of the Latin Hypercube and GAs. Moreover, the methodology can deal with both continuous and discrete variables, which allows not only the optimization of the material assignation, but also the optimization of continuous variables related to dimensions of the scaffold. On the other hand, the method can be applied both in solid and in hierarchical-based geometries since the concept is the same. In the case of solid geometries, an important database of mechanical properties over degradation time is required for different materials and printing configurations (infill density, infill strategy, layer height, etc.). Moreover, the methodology also has potential to be combined with novel modelling techniques based on element activation, although further research should be accomplished for this purpose.

7. ACKNOWLEDGEMENTS

This research was funded by BioAM project (Improvement of the biofunctionality of polymeric scaffolds obtained by additive manufacturing, DPI2017-88465-R) and supported by BAMOS project (Biomaterials and Additive Manufacturing: Osteochondral Scaffold innovation applied to osteoarthritis, H2020-MSCA-RISE-2016-734156).

References

1. Mota C, Puppi D, Chiellini F, Chiellini E. Additive manufacturing techniques for the production of tissue engineering constructs. *Journal of Tissue Engineering and Regenerative Medicine* 2015; **9**(3):174–190. doi:10.1002/term.1635.

2. O'Brien FJ. Biomaterials & scaffolds for tissue engineering. *Materials Today* 2011; **14**(3):88–95. doi:10.1016/S1369-7021(11)70058-X.

3. Hutmacher DW. Scaffolds in tissue engineering bone and cartilage. In: Williams DF, ed. *The Biomaterials: Silver Jubilee Compendium*. Oxford: Elsevier Science; 2000:175–189.

4. Murphy CM, Haugh MG, O'Brien FJ. The effect of mean pore size on cell attachment, proliferation and migration in collagen–glycosaminoglycan scaffolds for bone tissue engineering. *Biomaterials* 2010; **31**(3):461–466. doi:10.1016/j.biomaterials.2009.09.063.

5. Van Bael S, Chai YC, Truscello S, et al. The effect of pore geometry on the in vitro biological behavior of human periosteum-derived cells seeded on selective laser-melted Ti6Al4V bone scaffolds. *Acta Biomaterialia* 2012; **8**(7):2824–2834. doi:10.1016/j.actbio.2012.04.001.

6. Sobral JM, Caridade SG, Sousa RA, Mano JF, Reis RL. Three-dimensional plotted scaffolds with controlled pore size gradients: Effect of scaffold geometry on mechanical performance and cell seeding efficiency. *Acta Biomaterialia* 2011; **7**(3):1009–1018. doi:10.1016/j.actbio.2010.11.003.

7. Mason BN, Califano JP, Reinhart-King CA. Matrix Stiffness: A Regulator of Cellular Behavior and Tissue Formation. In: Bhatia SK, ed. *Engineering Biomaterials for Regenerative Medicine*. New York, NY: Springer New York; 2012:19–37.

8. Chen G, Dong C, Yang L, Lv Y. 3D Scaffolds with Different Stiffness but the Same Microstructure for Bone Tissue Engineering. *ACS Applied Materials & Interfaces* 2015; **7**(29):15790–15802. doi:10.1021/acsami.5b02662.

9. Park JS, Chu JS, Tsou AD, et al. The effect of matrix stiffness on the differentiation of mesenchymal stem cells in response to TGF-β. *Biomaterials* 2011; **32**(16):3921–3930. doi:10.1016/j.biomaterials.2011.02.019.

10. Breuls RGM, Jiya TU, Smit TH. Scaffold Stiffness Influences Cell Behavior: Opportunities for Skeletal Tissue Engineering. *The Open Orthopaedics Journal* 2008; **2**(1):103–109. doi:10.2174/1874325000802010103.

11. Solon J, Levental I, Sengupta K, Georges PC, Janmey PA. Fibroblast Adaptation and Stiffness Matching to Soft Elastic Substrates. *Biophysical Journal* 2007; **93**(12):4453–4461. doi:10.1529/biophysj.106.101386.

12. Peltola SM, Melchels FPW, Grijpma DW, Kellomäki M. A review of rapid prototyping techniques for tissue engineering purposes. *Annals of Medicine* 2008; **40**(4):268–280. doi:10.1080/07853890701881788.

13. Moroni L, de Wijn JR, van Blitterswijk CA. 3D fiber-deposited scaffolds for tissue engineering: Influence of pores geometry and architecture on dynamic mechanical properties. *Biomaterials* 2006; **27**(7):974–985. doi:10.1016/j.biomaterials.2005.07.023.

14. Hollister SJ, Maddox RD, Taboas JM. Optimal design and fabrication of scaffolds to mimic tissue properties and satisfy biological constraints. *Biomaterials* 2002; **23**(20):4095–4103. doi:10.1016/S0142-9612(02)00148-5.

15. Lin CY, Kikuchi N, Hollister SJ. A novel method for biomaterial scaffold internal architecture design to match bone elastic properties with desired porosity. *Journal of Biomechanics* 2004; **37**(5):623–636. doi:10.1016/j.jbiomech.2003.09.029.

16. Bendsøe MP. Optimal shape design as a material distribution problem. *Structural optimization* 1989; **1**(4):193–202. doi:10.1007/BF01650949.

17. Jammalamadaka U, Tappa K. Recent Advances in Biomaterials for 3D Printing and Tissue Engineering. *Journal of Functional Biomaterials* 2018; **9**(1). doi:10.3390/jfb9010022.

18. Gleadall A, Visscher D, Yang J, Thomas D, Segal J. Review of additive manufactured tissue engineering scaffolds: relationship between geometry and performance. *Burns & Trauma* 2018; **6**(1). doi:10.1186/s41038-018-0121-4.

19. Heljak MK, Święszkowski W, Lam CXF, Hutmacher DW, Kurzydłowski KJ. Evolutionary design of bone scaffolds with reference to material selection: EVOLUTIONARY DESIGN OF BONE SCAFFOLDS. *International Journal for Numerical Methods in Biomedical Engineering* 2012; **28**(6–7):789–800. doi:10.1002/cnm.2487.

20. Giannitelli SM, Accoto D, Trombetta M, Rainer A. Current trends in the design of scaffolds for computer-aided tissue engineering. *Acta Biomaterialia* 2014; **10**(2):580–594. doi:10.1016/j.actbio.2013.10.024.

21. Paz R, Pei E, Monzón M, Ortega F, Suárez L. Lightweight parametric design optimization for 4D printed parts. *Integrated Computer-Aided Engineering* 2017; **24**(3):225–240. doi:10.3233/ICA-170543.

22. Lophaven SN, Nielsen HB, Søndergaard J. A Matlab Kriging Toolbox. *Technical University of Denmark, Kongens Lyngby, Technical Report No. IMM-TR-2002-12* 2002.http://www.imm.dtu.dk/~hbn/dace/dace.pdfAccessedJuly 18, 2014.

23. Glibovets NN, Gulayeva NM. A review of niching genetic algorithms for multimodal function optimization. *Cybernetics and Systems Analysis* 2013; **49**(6):815–820. doi:10.1007/s10559-013-9570-8.

24. Adachi T, Osako Y, Tanaka M, Hojo M, Hollister SJ. Framework for optimal design of porous scaffold microstructure by computational simulation of bone regeneration. *Biomaterials* 2006; **27**(21):3964–3972. doi:10.1016/j.biomaterials.2006.02.039.

25. Gleadall A, Ashcroft I, Segal J. VOLCO: A predictive model for 3D printed microarchitecture. *Additive Manufacturing* 2018; **21**:605–618. doi:10.1016/j.addma.2018.04.004.

26. Brenken B, Barocio E, Favaloro A, Kunc V, Pipes RB. Development and validation of extrusion deposition additive manufacturing process simulations. *Additive Manufacturing* 2019; **25**:218–226. doi:10.1016/j.addma.2018.10.041.