Evolutionary Multi-Objective Optimization for Mesh Simplification of 3D Open Models

B. Rosario Campomanes-Álvarez[a,*], Oscar Cordón[abc] and Sergio Damas[a]
[a]*European Centre for Soft Computing, Gonzalo Gutiérrez Quirós s/n, 33600, Mieres, Asturias, Spain*
[b]*Department of Computer Science and Artifical Intelligence, University of Granada, Periodista Daniel Saucedo Aranda s/n, 18014, Granada, Spain*
[c]*Research Center on Information and Communication Technologies (CITIC-UGR), University of Granada, Periodista Rafael Gómez 2, 18014, Granada, Spain*

Abstract. Polygonal surface models are typically used in three dimensional (3D) visualizations and simulations. They are obtained by laser scanners, computer vision systems or medical imaging devices to model highly detailed object surfaces. Surface mesh simplification aims to reduce the number of faces used in a 3D model while keeping the overall shape, boundaries and volume. In this work, we propose to deal with the 3D open model mesh simplification problem from an evolutionary multi-objective viewpoint. The quality of a solution is defined by two conflicting objectives: the accuracy and the simplicity of the model. We adapted the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) and the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) to tackle the problem. We compare their performance with two classic approaches and two single-objective implementations. The comparison has been carried out using six different datasets from six corresponding real-world objects. Experimental results have demonstrated that NSGA-II and MOEA/D performs similarly and obtain the best solutions for the studied problem.

Keywords: 3D modeling, mesh simplification, evolutionary multi-objective optimization, NSGA-II, MOEA/D

1. Introduction

Polygonal surface models are the representation of 3D visualizations and simulations. They are obtained by laser scanners, computer vision systems or medical imaging devices to model highly detailed object surfaces. These surface models are used in many different areas such as computer vision, computer-aided design, medicine, and topography [34, 5, 55, 31, 2].

Typically, a surface model consists of thousands of polygons. The size of the corresponding file usually causes long processing times [36]. Obtaining a reduced model with a smaller polygonal surface and a similar accuracy is a challenge in the area. *Surface mesh simplification* is the process that aims to reduce the number of polygons used in a surface while preserving the overall shape, volume, and boundaries as much as possible [12].

There are several techniques in order to simplify a mesh. Decimation [50], clustering [42], and energy function optimization [30] are the most popular and classic methods. Recently, some other proposals have been made considering alternative ways to simplify 3D surfaces based on advanced optimization techniques as evolutionary algorithms [18, 32, 62].

Regardless the approach followed, all the latter methods consider the mesh simplification problem as a single-objective optimization task, either in a direct or indirect way. Nevertheless, two main criteria can be identified measuring the quality of the reduced 3D model, namely its accuracy to approximate the original mesh and its associated size, measured in terms of the number of polygons composing it. These two goals are clearly conflicting in nature as the more complex a mesh is, the more accurate it will be, and *vice versa*.

* Corresponding author. E-mail: rosario.campomanes@softcomputing.es

Hence, formulating mesh simplification as a multi-objective optimization problem (MOP) [11, 13, 1] can arise as a promising and very novel alternative to tackle this complex 3D modeling task. In particular, multi-objective evolutionary algorithms (MOEAs) [13, 57] have largely demonstrated their capability to deal with many different kinds of MOPs in a very efficient way [11, 13, 15, 69]. This family of methods shows the important advantage of being able to provide the user with a Pareto set of non-dominated solutions with different trade-offs on the satisfaction of the optimized objectives in a single run. In the mesh simplification framework, this would mean obtaining several and diverse alternative reduced 3D models with different compromises between their accuracy and their complexity (number of triangles), thus allowing the user to select the most appropriate for his/her specific conditions.

Thus, the formulation of mesh simplification as a MOP and the comparison with classic methods is an interesting research line which has been studied in this work. Up to our knowledge, this is the first study where mesh simplification is tackled from a multi-objective viewpoint in the literature.

Our methodology is thus based on *the simplification of a 3D open model* by an evolutionary multi-objective algorithm. An open model refers to a surface with open ends. The problem is based on the location of a certain number of points in order to approximate a mesh as accurately as possible to the initial surface. It will consider two conflicting objectives, the accuracy and the simplicity of a simplified mesh. In a previous research [8], we proposed the use of the computationally fast and extended non-dominated sorting genetic algorithm (NSGA-II) [15] MOEA to tackle the mesh simplification problem as a proof of concept. In the current contribution, we aim to demonstrate the good performance of our methodology by: i) showing how any other MOEA can be considered, and ii) developing a deeper experimentation to validate it with respect to both some classical and single-objective evolutionary mesh simplification methods. To do so, we have used the multi-objective evolutionary algorithm based on decomposition (MOEA/D) proposed in [66], which decomposes a MOP into a number of scalar optimization sub-problems and optimizes them simultaneously. Moreover, we also analyzed the method developed by Huang et al. in [32], that uses a single-objective algorithm to simplify 3D facial meshes, as well as two classic approaches, edge collapse decimation with a quadric error metric [27, 50, 20, 59], and vertex clustering with topology preserving [42].

This work is structured as follows. Section 2 introduces a short survey regarding classic techniques for mesh simplification and some others that consider evolutionary algorithms for that task. Section 3 describes all the components of the proposed approach. Section 4 presents the performed experiments and the results obtained. Section 5 concludes the whole work.

2. State of the Art in Mesh Simplification

Many different mesh simplification approaches have been proposed in the specialized literature [12, 27]. They can be either local or global. The former methods simplify a mesh by the iterative use of some local operator. The latter are applied to the input mesh as a whole. The following two subsections review local and global families of methods respectively. Finally, subsection 2.3 presents the existing evolutionary approaches to the problem.

2.1. Incremental Methods Based on Local Updates

The methods in this group run the simplification process as a sequence of local updates. Each update reduces the mesh size and decreases the surface approximation accuracy.

The decimation method [50, 29, 23] can be classified into three different approaches according to the difference of the selected objects: removal of an edge, removal of a triangle, and removal of a vertex. The latter approach can be guided by a quadric error metric algorithm [20, 59] based on the iterative contraction of vertex pairs.

Another example of an iterative method is the energy function optimization approach [30, 46]. The mesh reduction is iteratively obtained by performing legal moves on mesh edges: collapsing, swapping, or splitting. The progressive meshes method [29] is an enhanced version of the splitting technique.

Other methods based on surface signal [46] are used to simplify meshes by mainly using textures and colors.

2.2. Non Incremental Global Methods

The non incremental global methods simplify the mesh as a whole. Among them, the coplanar method [28] uses a named detecting plane to determine whether a vertex is near enough.

The re-tiling method [60] starts with a polygonal surface and creates a triangulation of it with a user-specified number of vertices. The number of polygons shared by any given edge is the main restriction.

The clustering technique [42] is based on geometric closeness. It uses the cube or *octree* neighborhood structured to group nearby vertices into a cluster. For each cluster, the method generates a new representative vertex [22]. This method preserves the topology of the mesh.

Besides, the algorithms based on wavelets [21, 26] and the simplification using envelopes [14] provide tight error bounds on arbitrary triangulated meshes while allowing topological changes during the simplification.

2.3. Mesh Simplification Approaches Based on Evolutionary Algorithms

Computational intelligence techniques have dealt with a wide variety of problems ranging from operational cost optimization [51, 45, 4], engineering design [35, 40, 52, 10], to copyright protection and data authentication [58]. Evolutionary algorithms have been largely and successfully applied to many different computer graphics, computer vision, and image processing tasks [9, 7, 64, 65, 53, 43, 44, 67, 47, 25]. In particular, there are a few studies based on applying evolutionary computation to deal with the mesh simplification problem [18, 32, 62].

In [18], Fujiwara and Sawai tackled the problem of approximating a human facial surface by constructing a triangular mesh with a limited number of sample points. They developed a single-objective genetic algorithm that selects a given number of points from the whole dataset and considers a Delaunay triangulation to build the simplified 3D model.

Huang and Ho [32] proposed an evolutionary algorithm as an extension of Fujiwara and Sawai's proposal. The improvement is based on using the orthogonal array crossover (OAX) (see Section 3.3).

Finally, Xiandong et al. proposed a method of triangular mesh reduction based on a new concept called super-face and the use of a genetic algorithm in [62].

3. Evolutionary Mesh Simplification of 3D Open Models

In this section, we formulate the 3D open model mesh simplification problem. We first detail a proposal based on Huang and Ho's method [32] and then describe our multi-objective approaches to tackle this complex problem.

3.1. Problem Formulation

Let M be a scanned 3D open model (Figure 1). It is possible to reduce this polygonal surface to a two dimensional problem. A 3D surface of this kind can be represented by the function:

$$f : (x, y) \in \Re^2 \rightarrow z \in \Re \qquad (1)$$

We have defined the mesh simplification problem in the 2D space due to efficiency purposes. The algorithm works with 2D models, which store fewer points than a 3D surface[1].

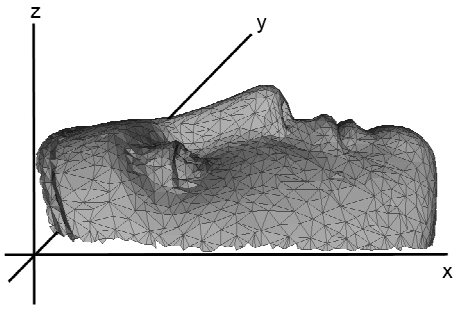

Figure 1. Surface representation in the standard Cartesian coordinate system.

Therefore, we need to locate n 2D points with n being less than N (the number of points of the original mesh). The number of points to be located can be either fixed a priori or automatically chosen by the algorithm. Our experiments are based on the latter option, i.e., the algorithm attempts to solve the problem with few data.

After locating the n points, a triangulation is performed using the new number of points in order to generate an approximate polygonal (triangular) mesh surface. In particular, methods described in Section 2.3 consider the Delaunay triangulation [3]. It is commonly used in problems such as the mesh generation process [63]. Given a set of points P in the plane, a Delaunay triangulation is a triangulation $D(P)$ such that no point in P is on the circumscribed

[1] Note that, in some concave surfaces or with high topological homotopy class, the 3D-2D mapping is restricted to some regions of the whole set [24]. However, this limitation is not present in our approach.

circle of any triangle of $D(P)$. Delaunay triangulations maximize the minimum angle of all angles of the triangle. They tend to avoid skinny triangles.

Let P_n be an initial set of points in $\Re^2$, its corresponding Delaunay triangulation is denoted by $D(P_n)$. In our problem, a chromosome encodes a configuration P_n of n points in the 2D space. The genotype space consists of those 2D configurations while the phenotype space includes the corresponding 3D models obtained using $D(P_n)$.

Let us have a population of N_{pop} individuals, each individual representing a certain mesh configuration. An individual is a simplified mesh, i.e., a mesh with fewer points than the reference model.

The members of the population share a global vector with the original mesh coordinates (original model). Each position of this structure stores the x, y and z coordinates. Every individual is defined by a binary chromosome with n genes *(the number of points which will be located on the new simplified mesh)*. A one value in the i_{th} position of the chromosome vector means that the i_{th} vertex of the original model remains in the simplified mesh represented by such chromosome. On the contrary, a zero means that there is not a point on the grid plane for this position. A graphical representation of the problem structures is shown in Figure 2.

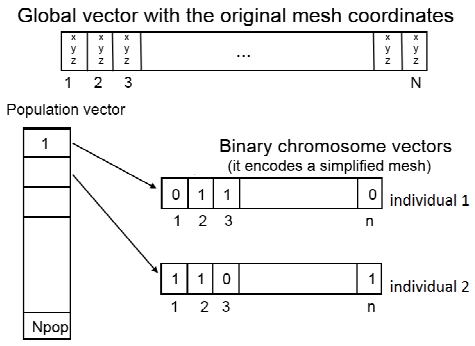

Figure 2. Problem structures scheme.

The four points of the mesh corners are included in every chromosome. This avoids an abnormal boundary shape thanks to the maintenance of the rectangular shape as a whole [18, 32].

The algorithm performs the following simplification process. First, it converts the original 3D model of N points (P_N) into a 2D model with the same number of points. Then, it carries out the simplification and obtains a new 2D mesh with n points (P_n), where n < N.

It applies Delaunay triangulation to the new 2D mesh getting $D(P_n)$. This 2D triangulated final mesh $D(P_n)$ is converted into 3D and thus a final 3D open model with n points and its corresponding number of triangles is obtained. The algorithm selects the model that best approximates the original one, i.e. the lowest error mesh. Figure 3 presents the scheme to obtain a simplified mesh from an original 3D open model by means of a binary-coded genetic algorithm.

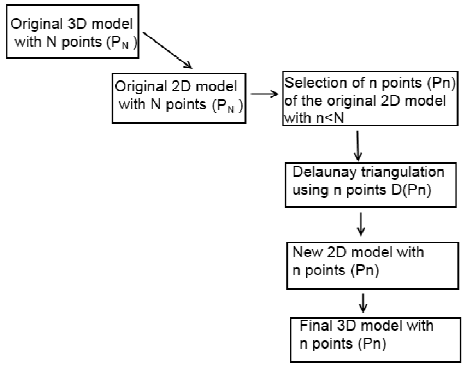

Figure 3. Brief scheme of the implemented method for mesh simplification.

3.2. Objectives to Be Optimized

We have considered two objectives to be jointly maximized, accuracy and simplicity. For a better formulation, the former is guided by the minimization of an error metric. The latter is given by the minimization of the number of triangles that compose the mesh (the lower the number of triangles is, the higher the model simplicity is). Therefore, we aim to minimize two objectives, the error and the number of triangles.

We have followed the same procedure presented by Huang et al. in [32] to calculate the approximation error, i.e., the error between the original and the approximated meshes. Each Delaunay triangle $T_i \in D(P_n)$ contains a certain number of mesh points (x, y). The distance d_p at each sample point p is defined as follows:

$$d_p = \left| z_p - \tilde{z}_p \right| , \quad (2)$$

where z_p is the height value of the surface at the point p and $\tilde{z}_p$ is the linearly interpolated value of height at p determined by the triplet of heights for the three vertices of triangle T_i.

The error e_i for T_i is the sum of these distances over all the sample points p inside T_i where:

$$e_i = \sum_{p \in T_i} d_p \tag{3}$$

The total approximation error e, defined by Eq. (4), is the sum of the errors e_i of all triangles that form the simplified mesh.

$$e = \sum_{T_i \in D(P_n)} e_i \tag{4}$$

3.3. Recombination

Recombination is the process in which a new individual solution is created from the information contained within two (or more) parent solutions. It is one of the most important operators in genetic algorithms. Recombination is randomly applied according to a crossover ranges in rate p_c, which typically ranges in [0.5, 1]. To apply recombination, two parents are selected and a random value u is generated uniformly. If u is lower than p_c, two offspring are created via recombination of the two parents. Otherwise, they are created by directly copying the parents [17].

We have used two variants for the crossover in our MOEAs: the classic uniform crossover [56] and the OAX recombination [41].

The uniform crossover [56] is based on dividing the parents into a number of sections of contiguous genes and reassembling them to produce offspring. It works by treating each gene independently and making a random choice as to which parent it should be inherited from. This is implemented by generating a string of L random variables from a uniform distribution over [0, 1]. In each position, if the value is below a parameter p (usually 0.5), the gene is inherited from the first parent; otherwise from the second. The second offspring is created using the inverse mapping.

The OAX recombination technique performed in [32] has also been considered in order to improve the performance of crossover. An efficient way to study the effect of several factors simultaneously is to use an orthogonal experimental design (OED) with orthogonal arrays (OAs) and factor analysis (FA). This kind of design is considered to provide the treatment settings at which one conducts the "all-factors-at-one" statistical experiments [41]. The orthogonal design defines some combinations for each experiment by using factor levels.

An OA is a matrix of numbers arranged in rows and columns where each row represents the levels of factors in each run and each column represents a specific factor. In the context of experimental matrices, orthogonal means statistically independent. FA can evaluate the effects of factors and determine the best level for each factor such that the evaluation is optimized. Using OAX the offspring chromosomes are formed from an intelligent combination of the good genes from their parents rather than the conventional random combination. The best of the two genes in the two parents is chosen by evaluating the contribution of individual genes to the fitness function based on OED.

Let there be α factors with two levels for each factor. The total number of experiments is 2^{α} for the popular "one-factor-at-a-time" study. The columns of two factors are orthogonal when the four pairs (1,1), (1,2), (2,1) and (2,2) occur equally frequently over all experiments. Generally, levels 1 and 2 of a factor represent selected genes from parents 1 and 2, respectively. To establish an OA of α factors with two levels, we obtain an integer $\beta = 2^{\lceil \log 2(\alpha+1) \rceil}$, build an orthogonal array $L_{\beta}(2^{\beta-1})$ with ß rows and ß-1 columns, use the first α columns, and ignore the other ß-α-1columns. The algorithm for constructing OAs can be found in [39]. An OED can reduce the number of experiments for FA. The number of OA experiments required to analyze all individual factors is only ß where $\alpha+1 \leq ß \leq 2\alpha$.

After proper tabulation of experimental results, the summarized data are analyzed using FA to determine the relative effects of levels of various factors. Let fe_t denote a fitness value, in our case fe_t is the total approximation error determined by Eq. (4), of the combination corresponding to the experiment t, where t= 1,…,ß. It defines the main effect of factor j with level k as S_{jk} where j=1, …, α and k=1, 2:

$$S_{jk} = \sum_{t=1}^{\beta} fe_t F_t, \tag{5}$$

where F_t= 1 if the level of factor j of experiment t is k; otherwise, F_t=0. In a minimization problem, the level 1 of factor j makes a better contribution to the fitness function than the level 2 of factor j does when $S_{j1} < S_{j2}$ (the opposite situation occurs in maximiza-

tion problems). The most effective factor j has the largest main effect different (MED) | S_{j1} - S_{j2}|.

3.4. NSGA-II

NSGA-II [15] is one of the most efficient and effective MOEAs following the elitist approach. Its particular fitness assignment scheme consists of sorting the population in different fronts with a non-domination order relation. Then, the algorithm combines the current population and its offspring generated with the standard bimodal crossover and polynomial operators to form the next generation. Finally, the best individuals according to non-dominance and diversity are chosen. This new version of NSGA [54] is characterized by a low computational complexity: O(NlogN), where N is the population size. The pseudo-code of the NSGA-II method [15] is outlined in the Appendix C of the supplementary information available at [73].

3.5. MOEA/D

A MOP can be stated as follows:

minimize $F(x) = (f_1(x),..., f_m(x))^T$,

subject to $x \in \Omega$, (6)

where Ω is the *decision space*, $F : \Omega \rightarrow R^m$ consists of *m* real-valued objective functions, and R^m is called the *objective space*.

MOEA/D [66] is a recent proposal of a MOEA based on explicitly decomposing the MOP showed in Eq.(6) into N scalar optimization sub-problems.

The algorithm solves these sub-problems simultaneously by evolving a population of solutions. At each generation, the population is comprised by the best solution found so far (i.e. since the start of the run of the algorithm) for each sub-problem. The neighborhood relations among these sub-problems are defined based on the distances between their aggregation coefficient vectors. The optimal solutions to two neighboring sub-problems should be very similar. Each sub-problem (i.e. scalar aggregation function) is optimized in MOEA/D by using information only from its neighboring sub-problems.

There are several approaches for converting the problem of approximation of a Pareto front into a number of scalar optimization problems. In the following, we introduce the Tchebycheff approach, which has been used in our experimental study due to the good results in terms of feasibility and efficiency obtained in [66].

Let $\lambda = (\lambda_1,..., \lambda_m)^W$ be a weight vector, and *m* be the number of sub-problems, i.e., $\lambda_i \geq 0$ for all i=1,…,*m* and $\sum_{i=1}^{m} \lambda_i = 1$ .The Tchebycheff approach considers a scalar optimization problem in the form:

minimize $g^{te}(x / \lambda, z^*) = \min_{1 \leq i \leq m} \{\lambda_i \left| f_i(x) - z_i^* \right|\}$

subject to $x \in \Omega$, (7)

where $z^* = (z_1^*,..., z_m^*)^W$ is the reference point. It is initialized as the lowest value of the objective function f_i found in the initial population.

For each Pareto optimal point x^* there exists a weight vector λ such that x^* is the optimal solution of Eq. (7) and each optimal solution of Eq. (7) is a Pareto optimal solution of the objective function. Therefore, the designer is able to obtain different Pareto optimal solutions by altering the weight vector.

3.6. Design of a Single-Objective Genetic Algorithm for Multi-Objective 3D Open Model Mesh Simplification

As a first approximation to the multi-objective problem, we will extend Huang and Ho's proposal [32] in order to compare it with our Pareto-based evolutionary approach.

This algorithm consists of several components such as population initialization, selection scheme, genetic operations, and termination criterion. As already introduced, each chromosome encodes a selection of 2D points P_n, having a phenotype given by its Delaunay triangulation $D(P_n)$. The fitness function to be minimized has been adapted in order to consider the two objectives introduced in Section 3.2:

$$F(m) = wE(m) + (1-w)T(m), \quad (8)$$

where *m* is the simplified mesh encoded in the chromosome, $D(P_n)$; $w \in [0, 1]$ is a weight, $E(m)$ is the error determined by Eq. (4) to be minimized, and $T(m)$ is the total number of triangles of *m*. The extension of the original method to tackle the fitness function in Eq. (8) is based on an evaluation function combining several objectives using a weighted sum [11, 13]. This method generates a set of Pareto optimal solutions by giving different weights to the function and running repeatedly the algorithm. The pseu-

do-code of the extended single-objective method is explained as follows:

Population initialization: The population is initialized by randomly locating n points to each individual. The four points of the 2D mesh corners are fixed to maintain the boundaries.

Selection scheme: It uses an elitist selection model in such a way that the individual with better fitness is always kept in the new population. It ranks N individuals according to this fitness given by Eq. (4). Then, it applies tournament selection [6].

Mutation: Each located point, coded at a gene in each individual, is randomly moved to one of the nearest neighbors on the vector with a probability p_m called the mutation rate. It makes mutation in each gene with a mutation rate equal to 1/*chromosome_length*.

Crossover: From two randomly chosen parents, we generate two offspring using two possible crossover operators, OAX [41] or uniform crossover [56] (see Section 3.3).

Termination criterion: To reach a maximum number of generations.

3.7. The NSGA-II Proposal

As said, two conflicting objectives are considered: accuracy and simplicity. Therefore, the two objectives to minimize are the error and the number of triangles of the mesh [8].

Given a mesh *m* for the multi-objective method, the fitness function is as follows:

$$\min \ FM^1(m) = E(m) \qquad (9)$$

$$\min \ FM^2(m) = T(m)$$

The method scheme is detailed below:

Population initialization: This procedure is the same than in the single-objective algorithm population initialization.

Selection scheme: The algorithm combines the current population with the obtained offspring using recombination in order to generate the next generation. The best individuals according to non-dominance and diversity are selected to be reproduced regarding the non-dominated fronts of NSGA-II (see Section 3.4).

Mutation: Each located point, coded at a gene in each individual, is randomly moved to one of the nearest neighbors on the vector with a probability p_m called the mutation rate. It makes mutation in each gene with a mutation rate equal to 1/*chromosome_length*.

Crossover: From two randomly chosen parents, we generate two offspring using two possible crossover operators, OAX [41] or uniform crossover [56].

Termination criterion: To reach a maximum number of generations.

3.8. The MOEA/D Proposal

We use the fitness function defined by Eq. (7). The pseudo-code of the method [66] is outlined as follows:

Initialization: Randomly generate *W* weight vectors λ. Compute the Euclidean distances between any two weight vectors and then work out the *W* closest weight vectors to each weight vector. For each $i = 1,2,...,N$ set $B(i) = \{i_1,...,i_W\}$, where $\lambda^{i_1},...,\lambda^{i_W}$ are the *W* closest weight vectors to λ^i.

Generate an initial population $x^1,...,x^N$ randomly. Set $FV^i = F(x^i)$.

Initialize $z = (z_1,...,z_m)^W$ by a problem-specific method. In our case, we have *m*=2 objectives to minimize, so *m*=2 sub-problems are to be minimized.

Update:

For $i = 1,...,N$ do

Reproduction: Randomly select two indexes *k*, *l* from $B(i)$, and then generate a new solution *y* from x^k and x^l by using genetic operators: The mutation is the same than in the NSGA-II proposal. The crossover consists of generating two offspring from two randomly chosen parents using uniform crossover [46].

Update of z: For each $j = 1,...,m,$ if $z_j < f_j(y')$, then set $z_j = f_j(y')$.

Update of Neighboring Solutions: For each index $j \in B(i),$ if $g^{te}(y'|\lambda^j, z) \le g^{te}(x^j|\lambda^j, z),$ then set $x^j = y'$ and $FV^j = F(y').$

Termination criterion: Stop if a maximum number of generations have been performed. Otherwise, go to *Update*.

4. Experiments

Six datasets have been considered to accomplish all the mesh simplification experiments. Three of those files correspond to synthetic meshes, Laurana.ply, Cheff.ply and Ramses.ply[2]. These models are provided courtesy of the AIM@SHAPE Shape Repository[3]. The rest of the datasets correspond to real-world models we are dealing with in some research projects within the forensic sciences area [49, 33]: two human skulls' meshes (Skull1.ply and Skull2.ply), given by the Physical Anthropology Laboratory of the University of Granada, Spain; and the mesh of the scanned Spanish historical monument face, *the lady of Elche* (FaceLadyElche.ply), kindly provided by the ITMA Materials Technology Centre in Asturias, Spain. All of them are 3D open models (see Figure 4).

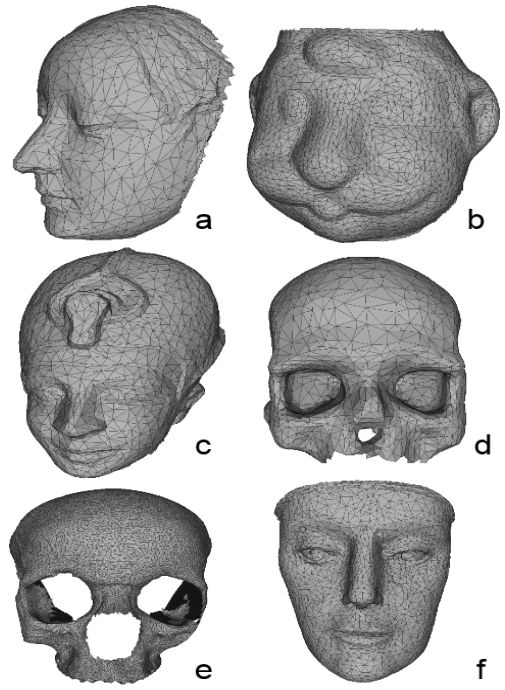

Figure 4. Problem meshes. (a) M1 dataset (Laurana.ply) 922 vertices, 1667 triangles. (b) M2 dataset (Cheff.ply) 2622 vertices, 4864 triangles. (c) M3 dataset (Ramses.ply) 1420 vertices, 2734 triangles. (d) M4 dataset (Skull.ply) 1055 vertices, 2004 triangles. (e) M5 dataset (Skull2.ply) 5196 vertices, 10000 triangles. (f) M6 dataset (FaceLadyElche.ply) 1777 vertices, 3254 triangles.

[2] The *ply* format is the polygon file format. It describes an object as a collection of vertices, faces and other elements, along with properties such as color and normal direction that can be attached to these elements.

[3] http://shapes.aim-at-shape.net/

4.1. Experimental Setup

The single-objective algorithm, NSGA-II, and MOEA/D have been implemented in C/C++ and all the experiments have been performed on an Intel Core 2 Quad CPU Q8400 2.66 GHz, with 4 GB RAM, running Windows 7 Professional.

We have distinguished two variants per dataset for the single-objective algorithm and NSGA-II. Variant 1 (v1) uses uniform crossover because it tends to produce more diversity than 2-point crossover, and variant 2 (v2) uses OAX with L4 matrix (based on Taguchi matrices proposed in [41]). The main reason why we selected OAX with a L4 matrix and not L8 or L12 is efficiency, as L4 took around 25 minutes per trial, so L8 and L12 are not tractable in terms of run times in practice. Respect to MOEA/D we have directly chosen the uniform crossover because it has a simple design and obtained better results than the OAX.

The used parameter values for the single-objective algorithms, NSGA-II and MOEA/D, have been the following: the population is set to be 100, 50 generations, crossover probability of 0.8, and mutation probability of 1/*chromosome_length*.

NSGA-II and MOEA/D have been run 10 times with different seeds. The initial populations are generated by uniformly randomly sampling from the feasible search space.

z^i in MOEA/D is initialized as the lowest value of f_i found in the initial population and the setting of weight vectors λ is the same as in [66]. *W* is set to be 20.

The previous parameters were selected after analyzing the performance of the algorithm proposals [15, 66], and some detailed sensitivity studies of genetic algorithm parameters [13, 16]. Both algorithms present a robust design that implies results less sensitive to parameter variation. Hence, this parameter setting could be used with other 3D inputs.

Regarding the single-objective algorithm, we considered different weight vectors. The weight of the first objective function, the total approximation error (Eq.(4)), ranges from 1 to 0 (step 0.1), and the simplicity (number of triangles) weights from 0 to 1 in the same step size. The algorithm has been run for each of the 11 weighted vectors so obtained.

4.2. The Classic Methods

We have also performed the mesh simplification process using two classic algorithms of different families of methods (incremental and non-incremental) in order to compare their results with our evolutionary proposals. The two chosen methods are edge collapse decimation based on quadric error metric [27, 50, 20, 59] and vertex clustering with topology preserving [42]. They are representative techniques among various surface based algorithms [68]. The edge collapse decimation has been selected because it leads to higher defined meshes and provides reasonable efficiency, while the vertex clustering is usually very robust and can be fast [27].

We developed the experiments by using the Meshlab software [72], which provides some tools and filters to apply both simplification techniques.

We have carried out the following procedure in order to run the classic methods: let us have an original mesh to be simplified using the quadric edge collapse decimation and the vertex clustering techniques. Ten different simplifications are performed changing the reduction percentage, i.e., from 5% to 50% of reduction with a step size of 5%. A specific reduction percentage value means that the algorithm reduces the mesh into a certain number of faces or triangles. After obtaining the simplified mesh with a specified number of triangles, we calculated the error between the new mesh and the original one in the way described in Section 3.2. Therefore, we will have ten solutions per algorithm that will be compared with those obtained by the single-objective algorithm, NSGA-II, and MOEA/D.

4.3. Performance Comparison: Pareto Fronts

The true Pareto-optimal front is usually considered to compare the performance among multi-objective algorithms. However, this front cannot be calculated in reasonable time in many real-world problems. That is the case of this study.

Hence, we have considered an approximation to the true Pareto front approximation (called pseudo-optimal Pareto front), which is obtained from the aggregation of the set of solutions P produced by every method in all the runs performed.

We calculated the Pareto front approximation of the single-objective algorithm as follows: we first merged the 11 solutions obtained by each weighted vector. Repeated solutions are removed. We finally produced the Pareto front approximation by performing a later domination check according to the Pareto dominance definition (the interested reader is referred to a brief review on MOP in Appendix A of the supplementary information available at [73]).

The Pareto front approximations for NSGA-II and MOEA/D have been calculated by merging the solutions obtained in the ten runs. Then, the repeated solutions are removed and the Pareto dominance is applied in order to get the final Pareto front approximation.

Finally, in the case of the two classic methods, the Pareto front approximation is calculated joining the ten calculated solutions in the way shown in Section 4.2.

4.4. Metrics of Performance: Quality Indicators

Multi-objective quality indicators represent a means to measure quality differences between Pareto front approximations on the basis of additional preference information. It is possible to check whether an algorithm provides significantly better approximation sets than another with respect to the preferences represented by the considered indicator [70, 37, 38].

We have used three of the most extended multi-objective performance indicators: the unary hypervolume indicator (HVR) [70], the binary epsilon indicator ($I\varepsilon$) [71], and the binary coverage metric (C) [70] (see Appendix B of the supplementary information at [73] for their detailed formulation). In addition, the cardinality of the Pareto set approximations obtained, i.e., the number of solutions composing them, will also be reported and analyzed.

4.5. Analysis of the Results

We have performed experiments using the six mentioned datasets. Figures 5 and 6 show the aggregated Pareto front approximations (see Section 4.3) for two of those datasets, M1 and M2, for illustration purposes (the Pareto front approximations of the rest of the datasets are available at [73]).

The Pareto front approximations of the classic approaches are not shown because they obtain higher errors than the rest of the methods. Their Pareto front approximations are far from the other five algorithms in the graphical representation.

The said figures show that the multi-objective methods have a better convergence than the single-objective ones. The NSGA-II proposals obtain solutions covering the whole space and achieve a greater

diversity. In general, MOEA/D and NSGA-II obtain solutions that dominate those achieved by the remaining methods in every dataset.

The single-objective techniques just tend to find solutions in a specific region of the search space. That is a typical behavior for weighted combination-based algorithms where specific weight vectors bias the search directions in different runs avoiding the obtaining of well spread Pareto fronts [13].

Table 1 presents the mean values for the C indicator of the two best algorithms, NSGA-II v1 and MOEA/D. This table reveals that in terms of the coverage metric, the final solutions achieved by NSGA-II v1 are better than MOEA/D for the M1, M3, and M5 datasets. Otherwise, MOEA/D outperforms NSGA-II for the M2, M4, and M6 instances.

Table 2 compares NSGA-II with MOEA/D by using the mean and the standard deviation values for the epsilon metric. In this case, NSGA-II v1 is better than MOEA/D for all the datasets but M2, although slight differences are always found. This shows the similar performance of both methods.

Figure 7 shows the mean values for the HVR metric of all the considered algorithms in the studied datasets. The figure reveals that NSGA-II v1 and MOEA/D outperform the rest of the methods in all datasets but M1.

Figure 8 presents the evolution of the cardinality of all the algorithms in the problem instances. NSGA-II achieves more diversity than MOEA/D in this study, being useful to find good compromises or trade-offs within the search space of our problem.

The single-objective algorithms can only detect one optimal solution (in our case, pseudo-optimal solution) in a single run while the multi-objective algorithms obtain a whole set of optimal (pseudo-optimal) solutions. So, multiple single-objective method runs are needed to achieve the same level of information that can be obtained from a single multi-objective method run, thus showing the capabilities of our proposal for the 3D open model mesh simplification problem solving.

The classic approaches obtain the highest errors in this study. The edge collapse decimation strategy achieves a good approximation and preserves the topology of the original mesh but at the cost of using a high number of triangles. The effect of the decimation is small and highly localized, so it tends to simplify the mesh by regions. A similar behavior corresponds to vertex clustering with topology preserving. We would need many simplifications of the original mesh to achieve a good relationship between accuracy and complexity in comparison with evolutionary algorithms. Hence, the higher the number of simplifications in the original mesh will be, the higher the total approximation error will be.

From the above results, we can conclude that both NSGA-II v1 and MOEA/D outperform the rest of the methods in the studied datasets, i.e., they achieve a better trade-off between the complexity and the accuracy of the resulting meshes.

Regarding the run time, all methods show a similar behavior but MOEA/D. The latter method needs less CPU time than the others. It spends between 6 and 8 minutes per run. The rest of the evolutionary algorithms take around 10-15 minutes per run for variant 1, a similar run time than the classic methods, and 20 minutes per run in variant 2. The interested reader is referred to a brief review on computational costs between NSGA-II and MOEA/D at [66].

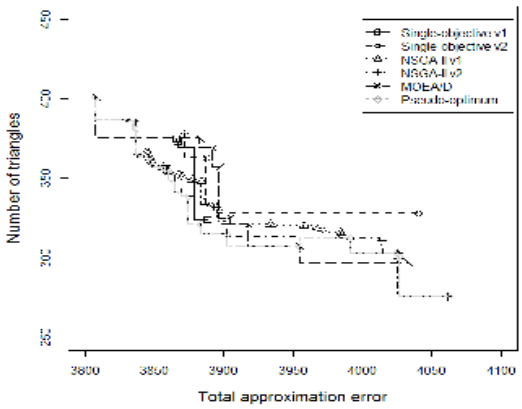

Figure 5. Pareto front approximations of the evolutionary algorithms for the M1 dataset.

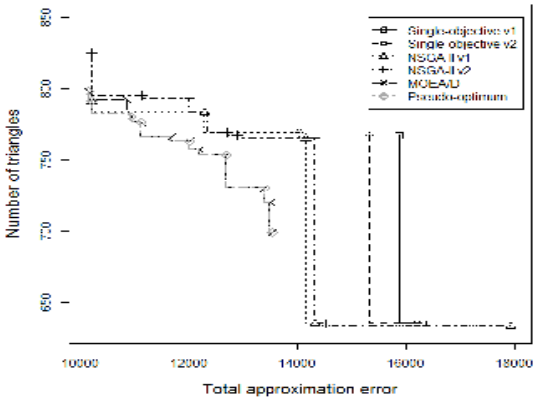

Figure 6. Pareto front approximations of the evolutionary algorithms for the M2 dataset.

Table 1

Mean values for the binary C indicator (Significant bold values treated as best result)

Dataset	C(NSGA-II v1,MOEA/D)	C(MOEA/D,NSGA-II v1)
M1	**0.57**	0.34
M2	0.38	**0.50**
M3	**0.59**	0.21
M4	0.42	**0.48**
M5	**0.28**	0.16
M6	0.37	**0.56**

Table 2

Mean and standard deviation for the binary Epsilon indicator (Significant bold values treated as best result)

Dataset	NSGA-II v1	MOEA/D
M1	**1.016 (0.0714)**	1.019 (0.0108)
M2	1.045 (0.0459)	**0.991 (0.0010)**
M3	**1.061 (0.0122)**	1.104 (0.0298)
M4	**1.039 (0.0032)**	1.256 (0.0376)
M5	**1.041 (0.0098)**	1.051 (0.0218)
M6	**1.031 (0.0126)**	1.053 (0.,0105)

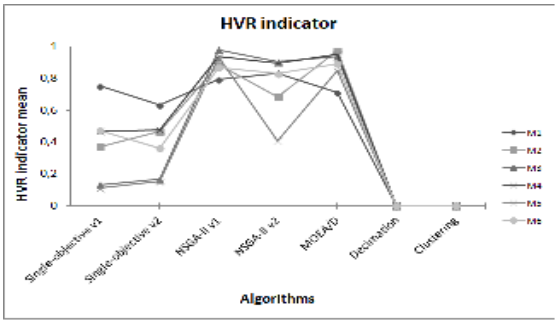

Figure 7. Evolution of the mean values of the HVR indicator in all the algorithms for all the datasets.

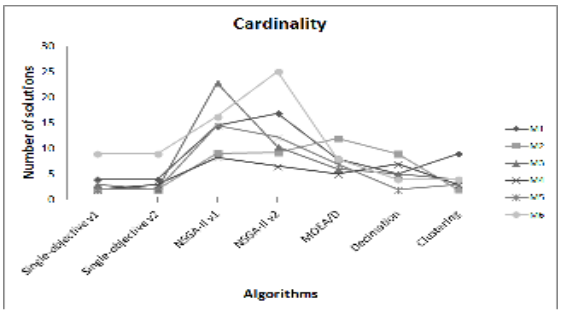

Figure 8. Evolution of the cardinality in all the algorithms for all the datasets.

4.6. The Wilcoxon Test

We have performed a Wilcoxon signed-rank test [61] to analyze the significance of the results in the comparison of the quality of the Pareto front approximations obtained by the single-, multi-objective, and classic algorithms by means of the previously explained unary and binary indicators. This is done in order to avoid the fact that one exceptionally good result in any of the compared algorithms produces a wrong analysis. Unlike the commonly used t-test, the Wilcoxon test does not assume normality of the samples and it has already demonstrated to be helpful analyzing the behavior of evolutionary algorithms [19]. Nevertheless, we should remark the fact that there is not any reference methodology to apply a statistical test to a binary indicator in multi-objective optimization. Thus, we have decided to follow the procedure described in [48]. The significance level considered in the performed test is p=0.05 for the I_ε indicator. For the C metric we chose a threshold of 0.75.

In view of this statistical study, available at [73], we can draw the conclusion that the multi-objective algorithms are significantly better in behavior than the rest of the methods. On the contrary, the two classic approaches perform significantly worse than the evolutionary ones with the applied significance level.

Specifically, the analysis reveals that NSGA-II v1 and MOEA/D are significantly better, in terms of accuracy and complexity, than the rest of the techniques for the C and epsilon indicators. Meanwhile, the results reveal that there are no differences among the solutions obtained by NSGA-II v1 and MOEA/D with the considered significance level.

Regarding the single-objective methods, the solutions achieved by variant 2 have a better tradeoff than the results obtained by variant 1 in the two metrics. Finally, the test does not obtain significant differences between the performances of the two classic algorithms.

4.7. Analysis and Comparison of Some Selected Solutions

We have selected three different solutions from each Pareto set approximation in order to evaluate the quality of the solutions obtained. Namely, the one with the best value in the first objective (minimum error solution), the one with the best value in the second objective (minimum number of triangles solu-

tion), and a compromise solution with the best trade-off value (best trade-off solution). The trade-off solution is selected as follows. We compute 1000 random weights $w \in [0, 1]$ and take the average value of the aggregation function of both objectives *Obj1* (error) and *Obj2* (number of triangles):

$$\overline{F}(s_i) = \frac{\sum_{i=j}^{1000} w_j Obj1(s_i) + (1 - w_j) Obj2(s_i)}{1000} \quad (9)$$

Due to the fact that the objectives *Obj1* and *Obj2* are not normalized we need to apply a factor in order to scale them:

$$\alpha = \frac{\sum_{i}^{|PA_a|} \frac{Obj2(s_i)}{Obj1(s_i)}}{|PA_a|}, \quad (10)$$

where PA_a is the cardinality of the Pareto front approximation and s_i is a solution of this Pareto front.

The final aggregation formula to compute the average value is the following:

$$\overline{F}(s_i) = \frac{\sum_{i=j}^{1000} \alpha w_j Obj1(s_i) + (1 - w_j) Obj2(s_i)}{1000} \quad (11)$$

The solution with the lowest aggregated value is selected. For each of the three chosen solutions, we present the values of the two objectives, error and number of triangles.

Figure 9 contains the representations of the three best solutions obtained by the evolutionary methods for the proposed models. We have excluded the classic approaches because their representations are far from the rest of the algorithms in the graphics.

Comparing the solutions obtained by each algorithm in the M1 dataset, MOEA/D, NSGA-II v2, and NSGA-II v1 achieve the best results. MOEA/D obtains the best minimum error solution followed by NSGA-II v2. MOEA/D and NSGA-II v1 perform the best tradeoff solution, and the best minimum number of triangles solution corresponds to NSGA-II v2.

Regarding the M2 dataset, NSGA-II v1 achieves the best results, although MOEA/D also obtains a good set of solutions.

Related to the M3 dataset, NSGA-II v1 performs the best minimum error solution. The latter algorithm and MOEA/D achieve similar results for the minimum number of triangles and the best trade-off solutions.

NSGA-II v1 and MOEA/D lead the three best solutions in the M4 and M6 datasets. We emphasize that NSGA-II v1, v2, and MOEA/D have a better performance than the other techniques for the M5 dataset.

For illustration purposes, figure 10 presents the best simplified 3D models achieved by NSGA-II v2 and MOEA/D tackling the M1 dataset. NSGA-II v2 obtains very good results in its minimum error solution (Figure 10b, 10g). This 3D mesh rightly approximates the contour of the face and other difficult regions such as nose, mouth, lips, forehead and eyes, obtaining a high quality approach comparing to the original model (Figure 10a, 10f) and (Figure 10b, 10g). However, the minimum number of triangles solution draws a worse shape of the face with respect to the original mesh, as expected (Figure 10a, 10c). Areas like the chin and right cheek have not been properly approximated (Figure 10c). The approximation of the forehead and the mouth is worse than the previously obtained solution as can be seen in Figure 10c, 10h.

Regarding the results of MOEA/D, the algorithm obtains a good contour of the face in the right side for both solutions (Figure 10d, 10e). The minimum error solution rightly approximates the mouth, nose, and eyes (Figure 10d, 10i). The accuracy of the simplified mesh showing the minimum number of triangles is good, but it approximates the nose worse than the previous solution (Figure 10e, 10j).

In general, NSGA-II and MOEA/D achieve some similar results (Figure 10b-e, 10g-j). They mainly differ in the approximation of the face contour (Figure 10b, 10d).

We have also analyzed the best solutions of the NSGA-II v1 and the clustering approach tackling the M6 dataset in [73] as supplementary information.

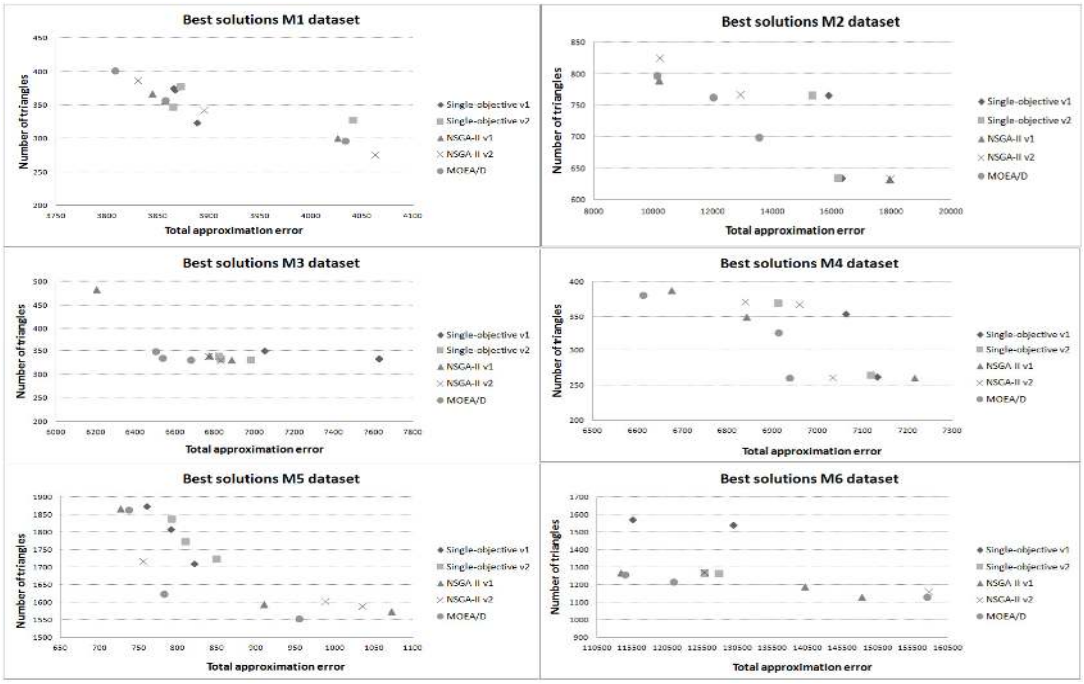

Figure 9. The best three selected solutions for each evolutionary algorithm in the analyzed datasets.

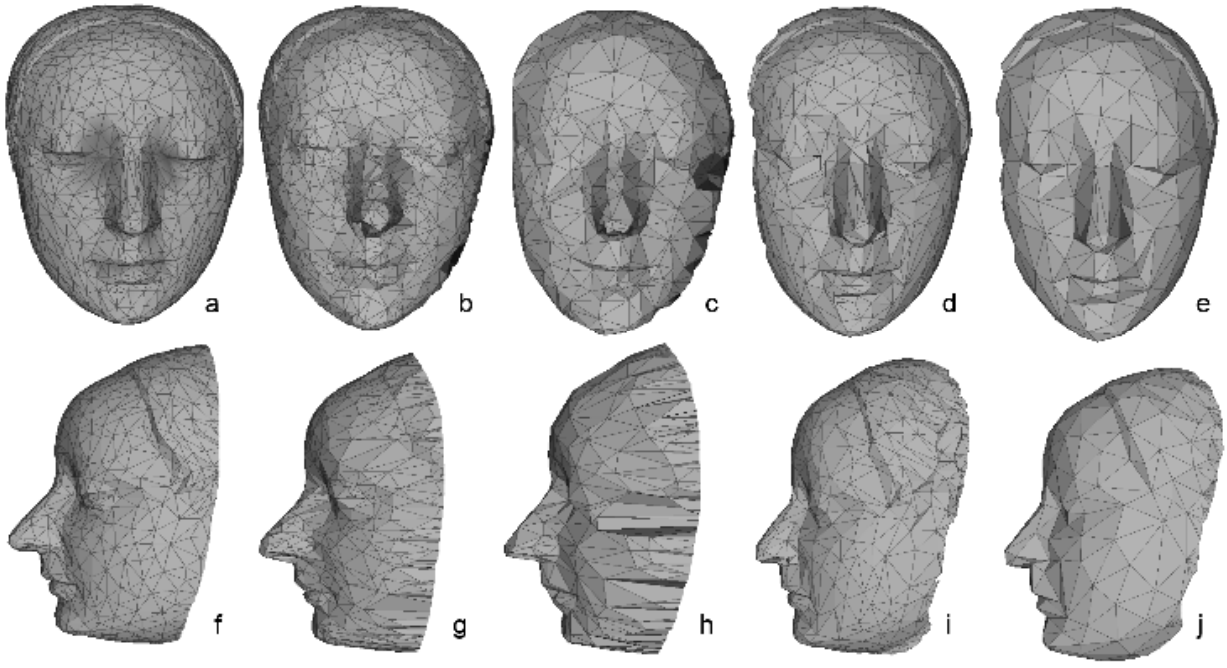

Figure 10. First row. M1 Dataset original model and solutions for NSGA-II v2 (b and c) and MOEA/D (d and e) algorithms, frontal views. (a) Original model, 1667 triangles. (b) Min. error solution, error=3829.92, triangles=387. (c) Min. number of triangles solution, error=4061.76, triangles=276. (d) Min. error MOEA/D solution, error=3808, triangles=401. (e) Min. number of triangles MOEA/D solution, error=4033, triangles=297. Second row. M1 Dataset original model and solutions for NSGA-II v2 (g and h) and MOEA/D (i and j) algorithms, lateral views. (f) Original model, 1667 triangles. (g) Min. error solution, error=3829.92, triangles=387. (h) Min. number of triangles solution, error=4061.76, triangles=276. (i) Min. error MOEA/D solution, error=3808, triangles=401. (j) Min. number of triangles MOEA/D solution, error=4033, triangles=297.

5. Concluding Remarks

We have proposed an evolutionary multi-objective framework to solve the 3D open model mesh simplification problem. The multi-objective approach has been implemented by using two specific MOEAs, NSGA-II and MOEA/D. They both have allowed us to find a set of solutions with different trade-offs between the accuracy and the simplicity of the simplified 3D open models.

In order to compare the performance of our proposals, we have considered three benchmarking methods, a single-objective evolutionary algorithm and two classic techniques. The experiments developed have been based on three publicly available datasets and three real-world 3D open models, one of them being a mesh of a scanned Spanish historical monument and the other two being human skull models from a forensic science research project. A Wilcoxon rank sum test has also been accomplished to analyze the significance of the obtained results.

From the analysis of those results, we can conclude that both the first NSGA-II variant proposed and MOEA/D obtain solutions that dominate those achieved by the rest of the methods. Their resulting meshes are more accurate, have a fewer number of triangles, and present a better trade-off between the two tackled objectives.

On the opposite, the classic approaches have obtained the highest modeling errors. They showed a particular behavior in the studied datasets. Many simplifications of the original mesh were required to achieve a good relationship between accuracy and complexity, in contrast to the considered evolutionary algorithms. Besides, the classic methods may excessively simplify a certain region, leaving other parts, which are expected to be simplified, untouched. Hence, the higher the number of simplifications in the original mesh is, the higher the total approximation error will be. Despite achieving a good precision in other studies [20, 42, 59, 68], this has not been our case. The reason could be that we considered meshes with open boundaries, where it is possible that large errors are generated in correspondence with the mesh boundary [12]. Nevertheless, both classic algorithms present reasonable processing times (similar to NSGA-II v1 and the single-objective v1, although larger than MOEA/D ones), an easier implementation design, and a good stability and robustness.

Besides the performance improvement, we should again remark that the multi-objective proposal allowed us to obtain a set of trade-off solutions in a single run of the algorithm. In this way, the decision maker can choose the most suitable solution (e.g. the most accurate, the simplest, or the one with the desired trade-off between both objectives) depending on the context the simplified 3D models will be used. This is a clear advantage over single-objective methods which try to find the "best" solution in a single run of the algorithm. Hence, more runs of the algorithm are needed to obtain a similar level of information to that achieved by the multi-objective algorithm, which implies a significantly higher computational effort.

Finally, as future works, it could be interesting to extend the MOEAs framework in order to tackle complete 3D surface approximations.

Acknowledgements

This work has been supported by the Spanish Ministerio de Educación y Ciencia under project TIN2009-07727 including European Development Regional Funds (EDRF) and the project SOCOVIFI2 (references TIN2012-38525-C02-01/TIN2012-38525-C02-02). The authors would like to thank the team of the Physical Anthropology Laboratory of the University of Granada and the ITMA Materials Technology Centre for providing us with real-world cases for our analysis.

References

[1] H. Adeli and S.L. Hung. Machine Learning - Neural Networks, Genetic Algorithms, and Fuzzy Sets, John Wiley and Sons, New York, 1995.

[2] H. Adeli and S. Kumar. Distributed Computer-Aided Engineering for Analysis, Design, and Visualization, CRC Press, Boca Raton, Florida, 1999.

[3] F. Aurenhammer. Voronoi Diagrams. A Survey of a Fundamental Geometric Data Structure, *ACM Computing Surveys*, 23(3), pp. 345-405, 1991.

[4] P. Baraldi, R. Canesi, E. Zio, R. Seraoui, and R. Chevalier. Genetic algorithm-based wrapper approach for grouping condition monitoring signals of nuclear power plant components, *Integrated Computer-Aided Engineering*, 18(3), pp. 221-234, 2011.

[5] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline, *Computer Graphics Forum*, 21(2), pp. 149-172, 2002.

[6] T. Bickle. Tournament selection. Handbook of Evolutionary Computation. T. Back, D.B. Fogel, Z. Michalewicz editors. IOP Publishing Ltd and Oxford University Press, pp. C2.3:1-4, 1997.

[7] S. Cagnoni, E. Lutton, and G. Olague. Editorial Introduction to the Special Issue on Evolutionary Computer Vision, *Evolutionary Computation* 16(4), pp. 437-438, 2008.
[8] B.R. Campomanes-Álvarez, S. Damas, and O. Cordón. Mesh Simplification for 3D Modeling using Evolutionary Multi-Objective Optimization, *Proceedings of the 2012 IEEE Congress on Evolutionary Computation,* pp.359-366, 2012.
[9] L. Carro-Calvo, S. Salcedo-Sanz, E. G. Ortiz-García, and A. Portilla-Figueras. An incremental-encoding evolutionary algorithm for color reduction in images, *Integrated Computer-Aided Engineering*, 17(3), pp. 261-269, 2010.
[10] T. Chabuk, J.A. Reggia, J. Lohn, and D. Linden. Causally-Guided Evolutionary Optimization and its Application to Antenna Array Design, *Integrated Computer-Aided Engineering*, 19(2), pp. 111-124, 2012.
[11] V. Chankong and Y. Y. Haimes. Multiobjective Decision Making Theory and Methodology, North-Holland, Amsterdam, 1983.
[12] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms, *Computers and Graphics*, 22(1), pp. 37-54, 1998.
[13] C.A. Coello, G. B. Lamont, and D. A. Van Veldhuizen. Evolutionary Algorithms for Solving Multiobjective Problems, Springer, Berlin, 2007.
[14] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright. Simplification envelopes, *Computer Graphics (SIGGRAPH '96 Proceedings)*, pp. 119-128, 1996.
[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2), pp. 182-194, 2002.
[16] K. A. De Jong. The Analysis of Behavior of a Class of Genetic Adaptive Systems. PhD Thesis, Department of Computer Science, University of Michigan, 1975.
[17] A.E Eiben and J.E Smith. Introduction to Evolutionary Computing, Springer, Berlin, Heidelberg, New York, 2003.
[18] Y. Fujiwara and H. Sawai. Evolutionary computation applied to mesh optimization of a 3-D facial image, *IEEE Transactions Evolutionary Computation*, 3(2), pp. 113-123, 1999.
[19] S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: a case of study, CEC'2005 Special Session on Real Parameter Optimization, *Journal of Heuristics* 15, pp. 617–644, 2009.
[20] M. Garland and P.S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics, *Computer Graphics (SIGGRAPH '97 Proceedings),* pp. 209–216, 1997.
[21] M.H. Gross, O.G. Staadt, and R. Gatti. Efficient triangular surface approximations using wavelets and quadtree data structures, *IEEE Transactions on Visualization and Computer Graphics,* 2(2), pp. 130-144, 1996.
[22] M.J. Haemer and M.J. Zyda. Simplification of objects rendered by polygonal approximation, *Computers and Graphics*, 15(2), pp. 175–184, 1991.
[23] B. Hamann. A data reduction scheme for triangulated surfaces, *Computer Aided Geometric Design*, 11(20), pp. 197–214, 2004.
[24] A. Hatcher. Algebraic Topology. Cambridge University Press, Cambridge, England. 2002.
[25] M. Hayashida, P. Ruan, and T. Akutsu. A Quadsection Algorithm for Grammar-Based Image Compression, *Integrated Computer-Aided Engineering*, 19(1), pp. 23-38, 2012.
[26] D.J. Hebert and H.-J. Kim. Image encoding with triangulation wavelets, *Proceedings SPIE*, 2569(1), pp. 381-392, 1995.
[27] P. Heckbert and M. Garland. Survey of Polygonal Surface Simplification Algorithms, Technical Report, Carnegie Mellon University, Pittsburg, 1997.
[28] P. Hinker and C. Hansen. Geometric optimization. Gregory M. Nielson and Dan Bergeron editors, *Proceedings Visualization '93*, pp. 189–195, 1993.
[29] G. Hoppe. Progressive meshes, *Computer Graphics (SIGGRAPH '96 Proceedings)*, pp. 99–108, 1996.
[30] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh Optimization, *Computer Graphics (SIGGRAPH '93Proceedings)*, pp. 19- 26, 1993.
[31] J. Hou, Z. Chen, X. Qin, and D. Zhang. Automatic image search based on improved feature descriptors and decision tree, *Integrated Computer-Aided Engineering*, 18(2), pp. 167-180, 2011.
[32] H.L. Huang, S. Y. Ho. Mesh optimization for surface approximation using an efficient coarse-to-fine evolutionary algorithm, *Pattern Recognition,* 36, pp. 1065-1081, 2003.
[33] O. Ibáñez, L. Ballerini, O. Cordón, S. Damas, and J. Santamaría. An experimental study on the applicability of evolutionary algorithms to craniofacial superimposition in forensic identification, *Information Sciences*, 179(3), pp. 3998-4028, 2009.
[34] K. Ikeuchi and Y. Sato. Modeling from Reality, Kluwer Academic Publishers Norwell, MA, USA, 2001.
[35] R. Jafarkhani and S.F. Masri. Finite Element Model Updating Using Evolutionary Strategy for Damage Detection, *Computer-Aided Civil and Infrastructure Engineering*, 26(3), pp. 207-224, 2011.
[36] S. Kim and C. Kim. Extracting feature lines from unstructured meshes using a model feature map, *Integrated Computer-Aided Engineering*, 13(2), pp. 101-112, 2006.
[37] J. Knowles and D. Corne. On metrics for comparing non-dominated sets, *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, 1, pp. 711–716, 2002.
[38] J. Knowles, L. Thiele, and E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2006.
[39] Y. Leung and Y. Wang. An orthogonal genetic algorithm with quantization for global numerical optimization, *IEEE Transactions on Evolutionary Computation* 5(1), pp. 41-53, 2001.
[40] G.C. Marano, G. Quaranta, and G. Monti. Modified genetic algorithm for the dynamic identification of structural systems using incomplete measurements, *Computer-Aided Civil and Infrastructure Engineering*, 26(2), pp. 92-110, 2011.
[41] T. Mori. Taguchi techniques for image and pattern developing technology, Prentice-Hall, Englewood Cliffs, N.J., 1995.
[42] G. Pengdong, L. Ameng, L. Yongquan, H. Jintao, L. Nan, and Y. Wenhua. Adaptative Mesh Simplification Using Vertex Clustering with Topology Preserving, *CSSE '08 Proceedings of the 2008 International Conference on Computer Science and Software Engineering,* 2, pp. 971-974, 2008.
[43] D. Plemenos and G. Miaoulis. (Eds.): Intelligent Computer Graphics 2011*, Studies in Computational Intelligence 374,* Springer-Verlag, Berlin, Heidelberg, 2012.
[44] D. Plemenos and G. Miaoulis. Intelligent Techniques for Computer Graphics, D. Plemenos, G. Miaoulis editors, Artificial Intelligence Techniques for Computer Graphics 159, pp. 1-14, Springer-Verlag, Berlin, Heidelberg, 2008.
[45] R. Putha, L. Quadrifoglio, and E. Zechman. Comparing Ant Colony Optimization and Genetic Algorithm Approaches for Solving Traffic Signal Coordination under Oversaturation Conditions, *Computer-Aided Civil and Infrastructure Engineering*, 27(1), pp. 14-28, 2012.

[46] P.L. Qu and G.W. Meyer. Perceptually Guided Polygon Reduction, *IEEE Transactions on Visualization and Computer Graphics*, 14(5), pp. 1015-1029, 2008.
[47] R. del Riego, J. Otero, and J. Ranilla. A low cost 3D Human Interface Device using GPU-based Optical Flow Algorithms, *Integrated Computer-Aided Engineering*, 18(4), pp. 391-400, 2011.
[48] L. Sánchez and J.R. Villar. Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms. *Information Sciences,* 178, pp. 950-970, 2008.
[49] J. Santamaría, O. Cordón, S. Damas, J.M. García-Torres, A. Quirin. Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. *Soft Computing*, 13 (8-9), pp. 883-904, 2009.
[50] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangular meshes, *Computer Graphics (SIGGRAPH '92 Proceedings)*, pp. 65–70, 1992.
[51] J. Sedano, A. Berzosa, J.R. Villar, E.S. Corchado, and E. de la Cal. Optimising operational costs using Soft Computing techniques, *Integrated Computer-Aided Engineering*, 18(4), pp. 313-325, 2011.
[52] L. Sgambi, K. Gkoumas, and F. Bontempi. Genetic Algorithms for the Dependability Assurance in the Design of a Long Span Suspension Bridge, *Computer-Aided Civil and Infrastructure Engineering*, 27(9), pp. 655-675, 2012.
[53] L. Silva, O.R.P. Bellon, and K.L. Boyer. Precision Range Image Registration using a Robust Surface Interpenetration Measure and Enhanced Genetic Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5), p.762-776, 2005.
[54] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms, *Evolutionary Computation*, 23, pp. 221-248, 1995.
[55] Y. Sun, J. Paik, A. Koschan, and M. Abidi. Surface modeling using multi-view range and color images, *Integrated Computer-Aided Engineering*, 10(1), pp. 37-50, 2003.
[56] G. Sywerda. Uniform crossover in genetic algorithms, *Proceedings of the Third international Conference on Genetic Algorithms* (George Mason University, United States). J. D. Schaffer, Ed. Morgan Kaufmann Publishers, San Francisco, CA, pp. 2-9, 1989.
[57] K. C. Tan, E. F. Khor, and T. H. Lee. Multiobjective Evolutionary Algorithms and Applications, Springer-Verlag, United Kingdom, 2005.
[58] H. Tao, J.M. Zain, M.M. Ahmed, A.N. Abdalla, and W. Jing. A Wavelet-Based Particle Swarm Optimization Algorithm for Digital Image Watermarking, *Integrated Computer-Aided Engineering*, 19(1), pp. 81-91, 2012.
[59] D.M. Thomas, P. K. Yalavarthy, D. Karkala, and V. Natarajan. Mesh Simplification Based on Edge Collapsing Could Improve Computational Efficiency in Near Infrared Optical Tomographic Imaging, *IEEE Journal of Selected Topics in Quantum Electronics*, 18(4), pp. 1493-1501, 2012.
[60] G. Turk. Re-tiling polygonal surfaces, *Computer Graphics (SIGGRAPH '92 Proceedings)*, pp. 55–64, 1992.
[61] F. Wilcoxon. Individual Comparisons by Ranking Methods, *Biometrics Bulletin*, 1(6), pp. 80-83, 1945.
[62] T. Xiaodong, W. Yuexian, Z. Xionghui, and R. Xueyu. Mesh Simplification Based on Super-Face and Genetic Algorithm, *Reverse Engineering. Advanced Manufacturing Technology*, 29, pp. 303-312, 2002.
[63] G. Yagawa, S. Yoshimura, and K. Nakao. Automatic mesh generation of complex geometries based on fuzzy knowledge processing and computational geometry, *Integrated Computer-Aided Engineering*, 2(4), pp. 265-280, 1995.
[64] E. Yeguas, R. Joan-Arinyo, and M. V. Luzón. Modeling the Performance of Evolutionary Algorithms on the Root Identification Problem: A Case Study with PBIL and CHC Algorithms, *Evolutionary Computation,* 19(1), pp. 107-135, 2011.
[65] M. Zhang. Improving object detection performance with Genetic Programming, *International Journal on Artificial Intelligence Tools*, 16, pp. 849-873, 2007.
[66] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, *IEEE Transactions on Evolutionary Computation*, 11(6), pp. 712-731, 2007.
[67] H. Zhou and Y. Liu. Accurate Integration of Multi-view Range Images Using K-Means Clustering, *Pattern Recognition*, 41(1), pp. 152-175, 2008.
[68] M. Zhou and M.Y. Wang. Engineered Model Simplification for Simulation Based Structural Design, *Computer-Aided Design and Applications*, 9(1), pp. 87-94, 2012.
[69] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, 3(4), pp. 257–271, 1999.
[70] E. Zitzler, L. Thiele, and K. Deb. Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, 8(2), pp. 173–195, 2000.
[71] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation*, 7(2), pp. 117–132, 2003.
[72] http://meshlab.sourceforge.net/ [Accessed: 2nd July 2013]
[73] Supplementary information. http://docs.softcomputing.es/public/afe/ICAE-on-line-supplement.pdf [Accessed: 2nd July 2013]