Struct Multidisc Optim (2008) 35:541–550
DOI 10.1007/s00158-007-0122-6

INDUSTRIAL APPLICATION

Structural optimization of as-built parts using reverse engineering and evolution strategies

Manuel J. García · Pierre Boulanger · Miguel Henao

Received: 2 October 2006 / Revised: 23 February 2007 / Published online: 14 June 2007

Abstract In industry, some parts are prone to failures or their design is simply sub-optimal. In those critical situations, one would like to be able to make changes to the part, making it lighter or improving its mechanical resistance. The problem of as-built parts is that the original computer-aided design (CAD) model is not available or is lost. To optimize them, a reverse engineering process is necessary to capture the shape and topology of the original design. This paper describes how to capture the original design geometry using a semi-automated reverse engineering process based on measurement provided by an optical 3D sensor. Following this reverse engineering process, a Fixed Grid Finite Element method and evolutionary algorithms are used to find the optimum shape that will minimize stress and weight. Several examples of industrial parts are presented. These examples show the advantages and disadvantages of the proposed method in an industrial scenario.

Presented at the 7th World Congress on Computational Mechanics, Los Angeles 2006.

M. J. García (✉)
EAFIT University, Cr 49 No. 7 sur 50, Medellin, Colombia
e-mail: mgarcia@eafit.edu.co

P. Boulanger
Department of Computing Science
University of Alberta
Edmonton, Canada
e-mail: pierreb@cs.ualberta.ca

Keywords Reverse engineering · Shape optimization · Fixed grid · Finite element analysis · Evolution strategies · Genetic algorithms

1 Introduction

In industry, some parts are prone to failures or their design is simply sub-optimal. One would like, for example, to be able to make a part lighter but with the same mechanical resistance as the original. Three-dimensional structural optimization is a well-known robust method aiming at improving a design in an optimal fashion by changing the dimensional properties of the original part. Many objective functions can be used to optimize a part. Typically, objective functions can be set to lower its weight, to increase its structural stiffness, and to reduce stress or weighted combinations of those constraints. In general, this problem is solved by computing the stress and strains of the original part using finite element methods (FEM) for some predefined conditions. Structural optimization is the process of changing the dimensional properties of this part to find the optimal design that minimizes this objective function. Many methods can be used to compute this optimal condition such as gradient decent or Levenberg–Marquardt algorithms (Levenberg 1944). Structural optimization has been widely investigated for the last three decades (Haftka and Gurdal 1992). A Survey of structural Optimization in multidisciplinary aerospace design is presented in (Sobieszczanski-Sobieski and Haftka 1996) and a survey applied to mechanical product development is presented in (Saitou et al. 2005). Because of the complexity of the optimization function with many local minima, stochastic optimization techniques are preferred to gradient-based techniques. Evolutionary algorithms have been used for optimization since their introduction 40 years ago. They were developed independently by research groups in the USA and Europe (Bäck and Schwefel 1996). They mimic Darwin's theory of evolution initially presented in his book *On the Origin of Species by Means of Natural Selection*, where an optimal population for a specific environment is selected by a process called "natural selection." Natural selection is the mechanism by which those individuals who are better adapted to a particular environment survive over generations, while the less adapted individuals are eliminated. Evolutionary computation approximates Darwin's theory in the form of an algorithm (Bäck 1996). Application to shape optimization can be found in Woon et al. (2000) and García and Gonzalez (2004).

The four main variants of evolutionary computation that have been established over the last years are genetic algorithms (Gen and Cheng 1997), evolution strategies, evolutionary programming, and genetic programming (Bäck 1996; Bäck and Schwefel 1996; Schwefel et al. 1995). These algorithms are applied to problems in which analytical information where the function's gradient is not available. All algorithms start with a population of individuals that represent a potential solution to the optimization problem. Probabilistic operators (mutation and recombination) are applied to the population of individuals to produce new generations of offsprings. Then, a selection process of individuals is performed to select the fittest individuals. The fitness of an individual is associated to an objective function that needs to be optimized. Evolution strategies have been applied before to the optimization of 2D structures (García and Gonzalez 2004). In which case, individuals of a population consist of a set or feasible geometries that satisfy some topological and mechanical constraints.

The problem with the structural optimization of as-built parts is that the original geometry of the design is lost. What needs to be done before any optimization process takes place is to capture the original design using 3D geometric sensors capable of measuring at high density and precision (the manufactured part). Following this measurement process, a reverse engineering (RV) process is necessary to transform the measured points into a parametric CAD model that can be optimized. Because structural optimization is a complex parameter space to explore, the CAD model produced by the RV process must have a minimal meaningful set of dimensional parameters that will be changed. In other works, some researchers (Spath et al. 2002) use complex geometric representations such as surface and volumetric meshes to do the same thing. However, these geometric models are not really suited for this optimization process, as the preservation of topology may be lost. It also assumes that the resulting optimized triangular mesh must be reverse engineered to capture the final optimal design.

Section 1 briefly describes the RV process with the structural optimization constraints in mind. Section 2 describes a general overview on how evolutionary algorithms can be used in the context of structural optimization, and Section 3 describes the geometric modeling scheme necessary to perform this optimization. Following the geometric reconstruction and parametrization, Section 4 describes how a fixed grid (FG) finite element analysis algorithm is used to compute stress and strain for some operating conditions at each instance of the new geometry. Using the results of this analysis, Section 5 describes the specifics of the evolutionary algorithm used in this implementation, and Section 6 describes how to modify the current geometry to optimize the design. Section 7 presents some experimental results obtained so far.

2 Reverse engineering for structural optimization

To be compatible with the shape optimization, the RV process needs to reconstruct a CAD compatible parametric model from multi-view 3D data produced by a geometric sensor such as a computed tomography (CT), or optical or sonic sensor. The choice of the sensing technology depends on the complexity of the part and the precision at which one digitize them. In this experiment, we used a Kreon line optical sensor that is based on optical triangulation to measure the distance

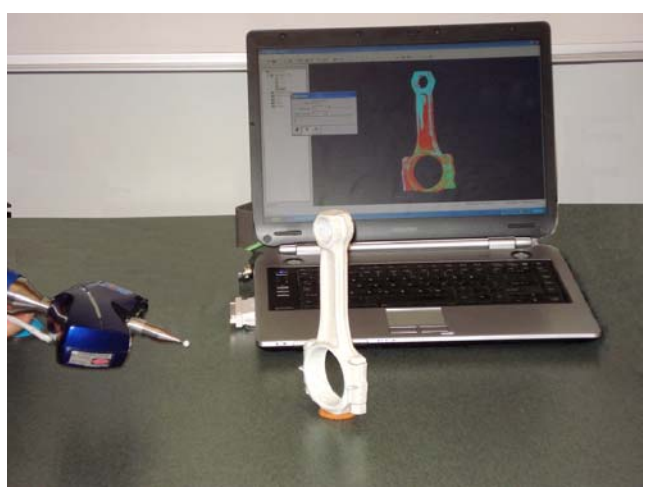


Fig. 1 The shape of the real part is acquired by an optical Kreon scanner mounted on a Faro arm

between the sensor and the surface of the object. The Kreon sensor used can measure at a rate of 30,000 pts/s with a resolution of 50 μm. Figure 1 shows a picture of the sensor during the digitization session of one of the optimized part.

As with most optical sensors, the part needs to be digitized from different points-of-view as illustrated on the computer screen in Fig. 1. This multi-view acquisition process is necessary to capture the whole surface of the part to deal with occlusions and specular reflections. Automatic multi-view registration and integration is a complex problem that is still subject to research. In many commercial systems, a practical solution using a mechanical arm is used to locate in real-time and at high precision the position of the scanning device. In our system, the Kreon scanner is mounted on the end effector of a positioning arm capable of a precision in the order of 15 μm. After a positioning process that computes the rigid transformation between the arm end affector position and the Kreon scanner, the combined precision of the measuring instrument is in the order of 67 μm at $\pm 3\sigma$.

In general, the raw data produced by those sensors are not directly suitable for import into CAD/CAM systems, as these range data consist of millions of single points located in the central coordinate system of the positioning arm and are called point clouds. To generate a CAD-compatible surface description from those point clouds, several data processing steps need to be performed: point cloud filtering, registration of all the point clouds into one common coordinate system, reconstruction from the global point cloud in a coherent triangular mesh that reconstructs the topology of the surface being digitized, and post-processing of the triangular mesh to fix holes and to sample the surface according to curvature. The final and most important step for shape optimization is to fit the triangulated surface using parametric modeling techniques such as nonuniform, rational B-splines (NURBS) in such a way that the final model can be optimized by modifying the parameters of the model. One can see in Figs. 2 and 3 two examples of such RV processes. Typically, even the simplest objects like the hook require at least five views to guaranty complete surface coverage. In the case of the connecting rod, we needed ten views on both sides to cover the whole surface. The two sides were integrated using a data-to-data registration process similar to the iterative closest-point algorithm or ICP (Besl and McKay 1992).

On the market, there are a lot of software capable of performing such processing steps. To name a few, the better ones are Polyworks™, Raindrop Geomagic™, and RapidForm™. Using Polyworks, the total digitizing time for both objects was approximately 15 min, and the processing time to transform the point cloud into a triangular mesh was approximately 1 h each. The resulting leak-free triangular models for each part was 20,000 triangles for the hook and 35,000 triangles for

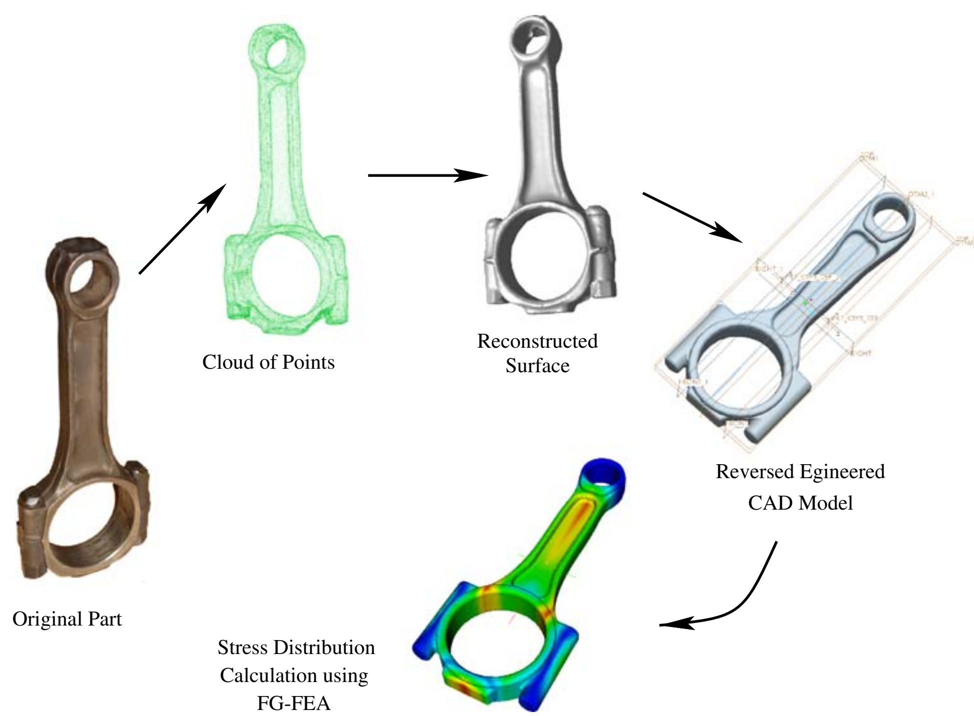


Fig. 2 From as-built part to FEM model (the connecting rod case): **a** picture of the real part, **b** digitized and integrated point cloud, **c** 3D triangulated meshed model of the point cloud, **d** extracted parametric nonuniform, rational B-splines (NURBS) model from the mesh model, and **e** stress distribution after FEM analysis on NURBS CAD model

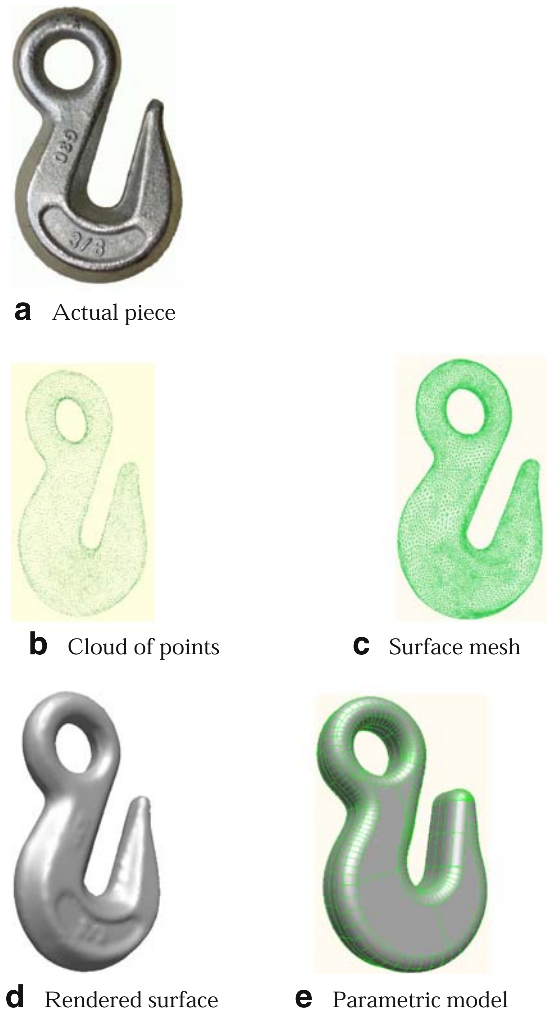


Fig. 3 Extraction of the CAD model from multi-view point cloud of a hook

the connecting rod. Using Polyworks' NURBS surface modeling tools, the triangulated model was then transformed into a parametric CAD model in 1 day. One of the requirements for shape optimization was that the final CAD had to be within a limited set of parameters that could be controlled by the shape optimization process. This final step of the modeling process is very difficult to automate, as it is very difficult to detect from the triangular mesh, the dominant topological shape of a part. In the future, one can imagine already parameterized generic shapes that can be adapted to the scanned object by a non-linear fitting algorithm.

3 FG evolution strategies for structural optimization

Evolution strategies structural optimization is a method developed for shape and parametric optimization of 3D linear elastic structures using evolution strategies. The main advantage of evolution strategies over genetic algorithms is the real valued representation of individuals instead of a binary encoding. There is no need for real-to-binary conversion, and the operators are based on real variable probabilistic operators rather than low-level manipulation of the bits. It is also independent of the type of coding used to store the variables, and finally, the genetic operator can be defined using state-of-the-art probabilistic and statistical operators (Bäck 1996).

In contrast with other structural optimization approaches, this method maintains and uses a high-level solid modeling representation of the structure. Optimization is accomplished by modifying the control points of the surfaces and CAD primitive parameters. Geometric operations are accomplished using Pro/Toolkit (PTC 1997), the Pro/Engineer™application programmer interface (API) that provides access to the geometric kernel functions. Also, it provides support for solid model representation and operation, and user interface.

Finally, fixed grid finite element analysis (FG-FEA) is used for structural analysis. It provides a fast estimate for the stress and displacement fields in the structure.

A structural optimization application was developed using the methods previously mentioned called provolution. It runs under the Pro/Engineer user interface. The architecture and description of the software can be found in (García et al. 2007; García 2005).

4 Geometric modeling

The geometric kernel allows solid manipulation via API. Boundary representation is obtained and expressed in form of NURBS. NURBS control points and also the solid primitives that can be maintained as design variables. The main advantage of this approach is that CAD information is maintained over the whole process, and therefore, a simple topology consistence test can be implemented to guarantee the geometric validity of the structures in the optimization process.

4.1 Non-uniform rational B-splines

The geometry of the structure is maintained by its boundary representation that contains information about the CAD primitives. Free-form shapes are represented by NURBS surfaces (Piegl and Tiller 1997). NURBS is a mathematical representation for

3–D surfaces, and it is supported by most of the CAD systems. A surface is represented as the following:

$$S(u,v)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij}N_{i,p}(u)N_{j,q}(v)P_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij}N_{i,p}(u)N_{j,q}(v)},$$

with u, v as parameters, $N_{i,p} and N_{j,q}$ as base functions, Pij as control points, and w_{ij} as weights.

The shape of the surface can be controlled by changing the position of the control points. See Fig. 4.

5 FG finite element analysis

FG finite element is a methodology to solve elasticity problems that was first introduced by García and Steven (1999) as an engine for numerical estimation of stress and displacement fields. The FG finite element method provides a tool for finding fast initial solutions to partial differential equation problems. The main idea is that the geometry is immersed in a grid of equal-sized elements, and the problem is solved using standard finite element analysis for a two-material problem. Special care is taken in the elements of the interface of the two materials to guarantee a minimum computational error.

Different material properties are given to the inside and to the outside. In general, the outside material is made of material with very low Young's modulus (like static air) and represent no interaction. The three different types of finite elements on a FG are shown in Fig. 5. The stiffness matrix for elements inside, I, or outside, O, can be calculated using a standard technique, while the stiffness matrix for elements on

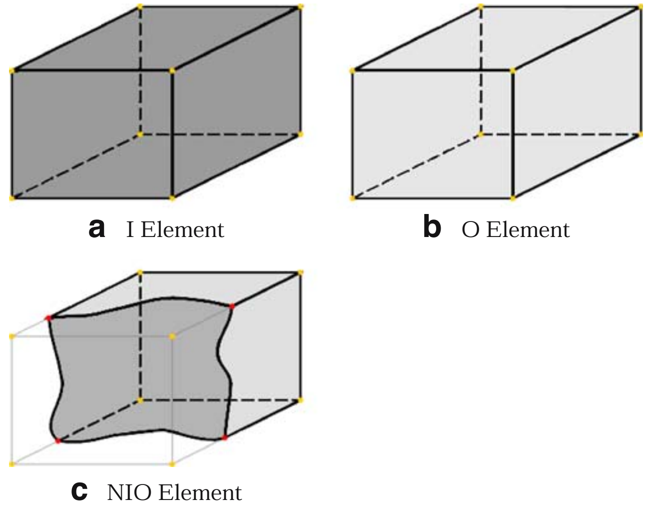


Fig. 5 Different types of finite elements in a FG method

the boundary, NIO, need a special treatment. Let E be Young's modulus and ν the Poisson's ratio for the constitutive material model; then, if I and 0 stand for inside and outside material, respectively, then the element stiffness matrix for an NIO element can be approximated by

$$\mathbf{K}_{NIO} = E_{NIO}\,\hat{\mathbf{K}}, \quad \text{with}$$
$$E_{NIO} = \xi\, E_I + (1-\xi)E_O\,. \tag{1}$$

The advantages of FG in structural optimization are numerous:

- It provides a fast analysis and re-analysis tool.
- No re-meshing is necessary, as only few positions of the stiffness matrix are changed when small changes in the geometry are made.
- The element stiffness matrix is constant. Therefore, the computation of the new analysis can be done with a minimum computational cost.
- Finally, there is no distortion of the elements of the grid.

Details on the theory and implementation of the FG method can be found in García and Steven (1999) for 2D cases and in Garcia et al. (2005) for 3D cases. It has been shown that a permissible error level is maintained with this approach and making this method suitable for structural optimization.

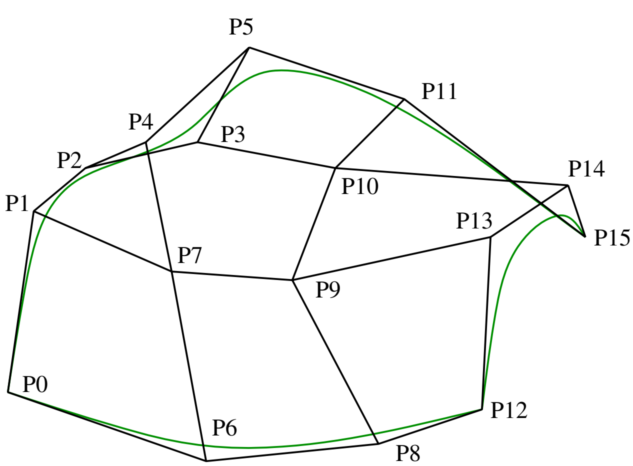


Fig. 4 NURBS representation of a surface

6 Evolution strategies

In evolution strategies, μ survivors are selected from the union of parents (μ) and offspring (λ) such that a course of evolution is guaranteed. An individual is composed of a pair of vector variables: $\vec{a} = (\vec{x}, \vec{\sigma})$, in which $\vec{a}$ is the individual, $\vec{x}$ is the vector of object variables, and $\vec{\sigma}$ is the vector of standard deviations used for mutation.

A high-level algorithm of the proposed evolution strategies can be summarized by:

```
t = 0
initialize P(t)
while (optimum limit is not reached) do
  Q(t) =recombine P(t)
  Q(t) =mutate Q(t)
  evaluate Q(t)
  P(t + 1) =select(P(t) ∪ Q(t))
  t = t + 1
end while
```

Where $P(t)$ is the parent population of μ individuals and $Q(t)$ is the offspring population of size $\lambda \geq \mu$. Evolution strategies were applied first by García and Gonzalez (2004) for 2D structures and by García (2005) for 3D structures. Detail of the evolutionary operators can be found in those references.

6.1 Structural optimization

Given an objective function $f : \mathbb{R}^n \to \mathbb{R}$, an individual is defined as $\vec{x} \in \mathbb{R}^n$ and is evaluated by $f(\vec{x})$. The vector $\vec{x}$ contains the parameters that define the structure. The goal is to find the fittest individuals $\vec{x}_f = \max(f(P_i(t)))$, where $P_i(t)$ represents the i^{th} individual of the parents population at the t generation:

$$f(\vec{x}_i) = w_1 \frac{\text{volume}(\vec{x}_i)}{\text{volume}(\vec{x}_0)} + w_2 \frac{\text{max\_disp}(\vec{x}_i)}{\text{max\_disp}(\vec{x}_0)} \quad (2)$$

subject to

$$\text{max\_vm}(\vec{\mathbf{x}}_i) \leq \text{max\_vm}(\vec{\mathbf{x}}_0).$$

The objective function as defined by 2 represents a point in the Pareto front determined by weights w_1 and w_2. Selection of the appropriate objective functions and the weights depends on the particular problem goals; e.g., under critical failure, it is more important to minimize the stress rather than to reduce the volume of the structure.

7 Optimization examples

7.1 Optimization of a connecting rod

This example consists of a connecting rod whose geometry was obtained by a RV procedure explained in previous sections. The final geometry of the rod and the control points used in the optimization procedure are shown in Fig. 6.

The loads applied in this case correspond to a compression in the direction of the main axis. The cylinder shape of the connectors is maintained constant during the optimization process. The control points are located in the connecting part and are restricted to maintain symmetry of the structure.

Figure 7 shows some results obtained with the provolution program. The first case (a) consisted of nine control points along the connecting rod as shown in Fig. 6. The optimization was stopped after 80 generations, and no improvement was observed in the total volume. In fact, it was observed that the optimization increased the volume by 7%. The final shape of the selected surface presented a wave form that implies that the optimization process was far from the optimum. This is probably due to the amount of control points, as well as the standard deviation chosen in the evolution strategy. In the second case (b), the number of control points was reduced by half. After 30 generations, it was observed that the optimization reduced the volume by 2% but increased the stress by 5%. In the third case (c), the standard deviation of the mutation operator was reduced by half. In this case, after 40 generations, there was no reduction in the volume (increased in 1%), but on the other hand, the maximum stress was reduced by 7%. In the last case (d), both the number of control points and the standard deviation were reduced by half. After 35 generations, the best results were obtained in this case where the volume was reduced by 3% and the stress reduced by 2%.

One should note that the connecting rod used in this example was selected because it is a well-known structural element in a motor. However, the connecting rod existed since the invention of the car, and over the time, its design has evolve into a very well-optimized structure.

Having said that, it is not surprising that the volume reduction obtained with the method presented in this paper is very low. Only small changes were observed

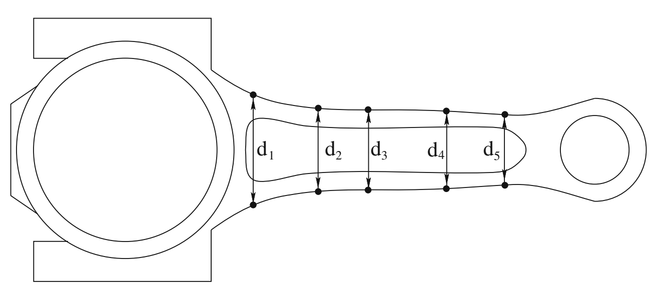


Fig. 6 Connecting rod and control points

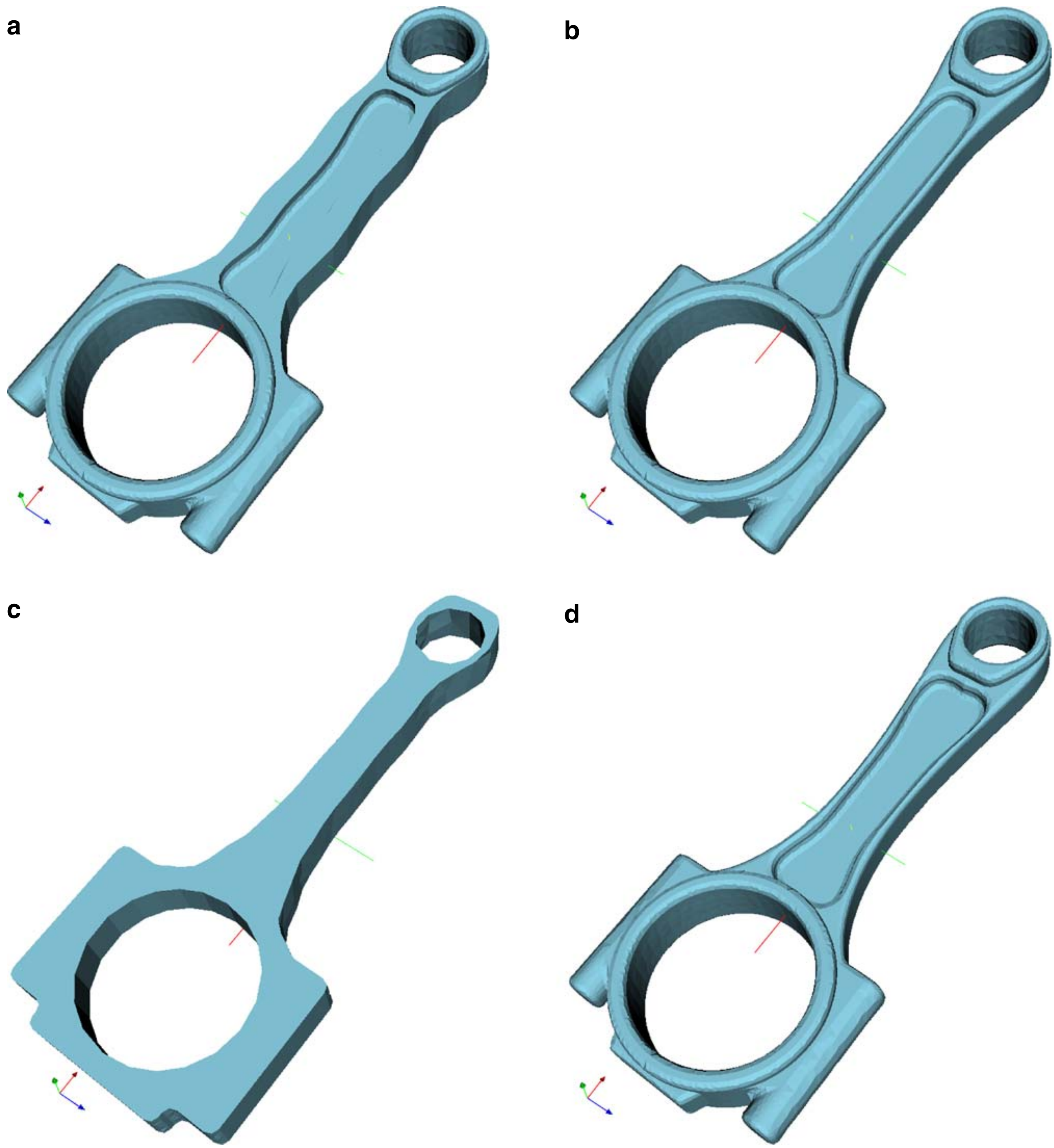
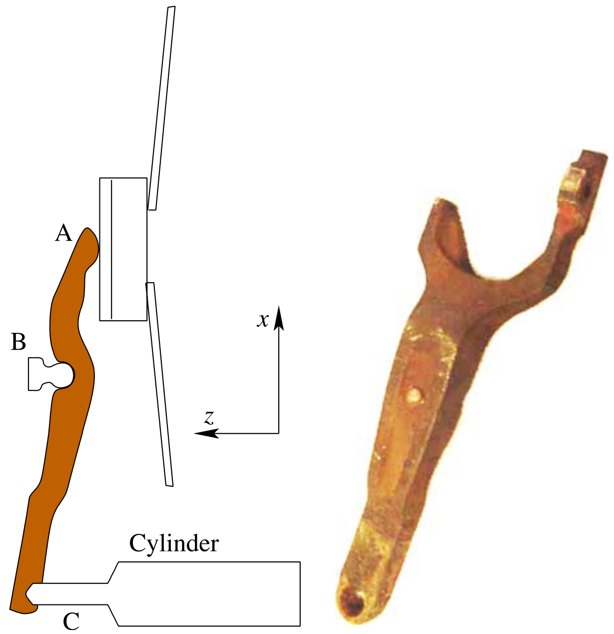


Fig. 7 Connecting rod results

in the shape, and in some cases, no improvement was obtained. Nevertheless, these results compared well with other attempts to optimize the same rod geometry as reported by Annicchiarico and Cerrolaza (2004). In their work, using a genetic algorithm technique and boundary element methods, they reduce the stress in the connecting rod at a cost of increasing the volume by 50%.

7.2 Optimization of a clutch fork

The clutch fork connects the slave cylinder to the clutch release bearing, commonly called the throw-out bearing. Figure 8a shows a sketch of a clutch system. The fork acts as a lever that is pivoted at point B (pivot ball). When the cylinder is actuated and the slave rod extends, pressure is applied to one end of the clutch



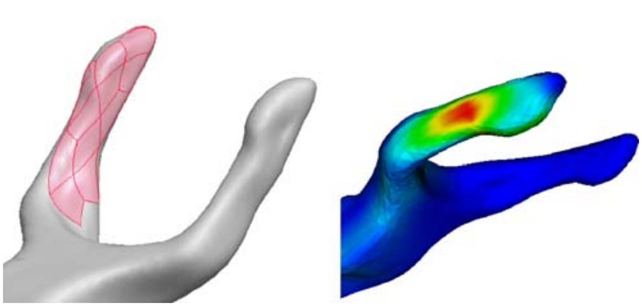
a Working diagram **b** A broken fork at one arm.

Fig. 8 Clutch fork. **a** A force is applied by the cylinder in *C* to displace the release bearing in *A*. **b** Photograph of a failed fork

fork C. This causes the fork to rotate on the pivot ball and slide the release bearing along the transmission input shaft into the clutch unit itself.

An original fork was digitized using the Kreon scanner, and a model was reconstructed with a total number of 43,000 triangles. This piece-wise linear surface was then transformed into a set of 150 NURBS patches using an automatic surfacing algorithm, each patch consisting of five control points. A detailed display of the NURBS patches at one of the arms is shown in Fig. 8a.

Due to its design shape, finite element analysis reveals that the region of maximum stress is located near the pivot. However, it has been found that many of them fail in one of the fork arms as shown in Fig. 8b.



a A patch of Nurbs surfaces. **b** Fixed Grid FEA.

Fig. 9 Detail of the surface modeling of the clutch fork and the analysis results for uneven boundary conditions

This is probably the result of uneven loads. In fact, a close inspection of scanned forks revealed that one of the arms of the fork can get in contact with the release bearing before the other. This is due to a fabrication defect that is very common in the cast iron process. To simulate this, the structure was modeled when only one arm is in contact against the release bearing. The boundary conditions correspond to a force in the z direction at point C, movement restriction at B in the z and y directions, and movement restriction at the contact point A in the z and x directions. FG finite element analysis was used, and the results are shown in Fig. 9b. It clearly reveals the point where the fork fails.

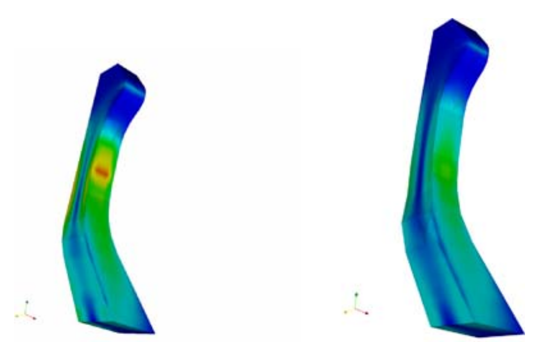
Furthermore, a structural optimization was performed on the fork model with one arm contact boundary condition. The objective function was chosen to be

$$f(\vec{x}_i) = w_1 \frac{\text{max\_vm}(\vec{x}_i)}{\text{max\_vm}(\vec{x}_0)} + w_2 \frac{\text{max\_disp}(\vec{x}_i)}{\text{max\_disp}(\vec{x}_0)} \quad (3)$$

subject to

$$\text{volume}(\vec{x}_i) \leq \text{volume}(\vec{x}_0).$$

This function minimizes stress with a volume constraint. Additionally, due to the complexity of the shape, few changes to the surface representation were made before optimization was applied. First, the definitions of the surface at the points of application of the boundary conditions had to be manually changed to properly mimic its working loads. Secondly, as the initial optimization test revealed, the amount of surfaces and control points was excessive, the convergence of the evolution strategy was very slow. Therefore, only local optimization of the critical part was possible to accomplish. An initial optimization gave the following



(a) Redefinition of the geometry representation (b) Optimsed shape

Fig. 10 Remodeled geometry for the fork optimization. Original and optimized shape. The *color* represent the von Mises stress. It can be noticed how the level of stress is decreased in the optimized shape results after 80 generations: a geometry with a stress reduction of 0.8% and an increase of the volume of 1.3%. The control points of the selected surfaces were used as design variables. Every run of the program took approximately 6 h and 18 min. The computation of the optimization parameters was accomplished by a trial-and-error process. This meant several days of computer–human interaction. In a search for improving this performance, a redefinition of the shape using a smaller number of surfaces (this is combining several ones into a single NURB) was performed. This reduced the number of control points and design variables. It also reduced the local oscillation of the shape due to random changes in the value of the control points produced by the evolutionary algorithm. The remodeling of shape geometry is presented in Fig. 10a. Figure 10b corresponds to the result of the optimization process after 40 generations with a population size of 13. In this case, the stress was reduced by 38.5%, and the volume was increased by 3.4%. The standard deviation was 1.5.

8 Conclusion

A procedure to perform shape optimization of as-built parts is presented in this article. The RV process starts with the acquisition of the geometry of the structures by using high-resolution laser scanners. The cloud of points produced in the scanning procedure is decimated, cleaned, meshed, and transformed into a parametric CAD model. After a de-featuring process of the parametric model (to reduce small geometric non-structural details), the number of NURBS is reduced, and some parametric CAD primitives and NURBS control points are identified and used as design variables. Selection of the control points plays an important role in the process. Moreover, as the number of control points increases, the performance of the method decreases. The search for the optimum shape is accomplished using evolution strategies.

This method serves as an exploration of the design space. Although, only two particular cases of the optimization function (or objective function) was shown in this paper, they can be constructed to satisfy particular demands such as to reduce stress concentration, to reduce weight, or, in general, to improve the mechanical response of the structure. The examples shown in this paper demonstrate the capabilities of the method. The experiments developed show that a good correlation of the optimization parameters will lead to a good solution. However, a bad choice of these parameters will delay the convergence of the method or will not converge at all. It seems that these parameters should be chosen for each particular problem. Also, it should be taken into account that the optimization of most of the reversed engineered structures aim to improve an already-designed object that, in many cases, is a good design. Therefore, the improvement obtained by the optimization seems to be small compared with the academic examples in which reduction of 50% of the volume is achieved.

Finally, alternative optimization methods should be used to enhance the performance of the process. It is well known than the combination of evolutionary strategies to search the local minimum plus the gradient-based method to refine the search are powerful combinations that take advantage of the two techniques. Other possibilities to improve performance include a parallel implementation of the FG and the evolution strategies methods. Nevertheless, this still requires the evaluation of a high number of feasible structures. In conclusion, shape optimization with evolution strategies (or genetic algorithms) seems to work fine in academic problems when the number of design variables is relatively small. However, in industrial problems, the number of trials to set the parameters plus the extended solution times make this method not the best choice. In fact, the results presented in this paper point more toward a gradient-based method or a combination of a stochastic and a gradient-based method.

References

Annicchiarico W, Cerrolaza M (2004) A 3-D boundary element optimization approach based on genetic algorithms and surface modelings. Eng Anal Bound Elem 28:1351–1361

Bäck T (1996) Evolutionary algorithms in theory and practice. Oxford University Pres, New York

Bäck T, Schwefel HP (1996) Evolutionary computation: an overview. In: IEEE International conference on evolutionary computation, Nagoya, Japan, pp 20–29

Besl P, McKay N (1992) A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach 14(2):239–256

Garcia M, Henao M, Ruiz O (2005) Fixed grid finite element analysis for 3D structural problems. Int J Comput Methods 2(4):569–586

García M, Henao M, Boulanger P (2007) Evolutionary algorithms applied to shape optimisation of 3-d structures. Int J Struct Multidiscip Optim (in press)

García M, Gonzalez C (2004) Shape optimisation of continuum structures via evolution strategies and fixed grid finite element analysis. Struct Multidisc Optim 26(1–2):92–98

García M, Steven GP (1999) Fixed grid finite elements in elasticity problems. Eng Comput 16(2):145–164

García MAH (2005) Evolutionary algorithms applied to shape optimisation of 3-D structures. Final year project, Mechanical Engineering, EAFIT University

Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley, New York

Haftka R, Gurdal Z (1992) Elements of structural optimization, 3rd edn. Kluwer, Norwell, MA

Levenberg K (1944) A method for the solution of certain problems in least squares. Q Appl Math 2:64–168

Piegl L, Tiller W (1997) The NURBS book, 2nd edn. Springer, Berlin

PTC (1997) Pro/Toolkit reference manual. Parametric Technologies, Waltham, MA

Saitou K, Izui K, Nishiwaki S, Papalambros P (2005) A survey of structural optimization in mechanical product development. J Comput Inf Sci Eng 5:214–226

Schwefel HP, Rudolph G, Bäck T (1995) Contemporary evolution strategies. In: European conference on artificial life, Granada, Spain, pp 893–907

Sobieszczanski-Sobieski J, Haftka RT (1996) Multidisciplinary aerospace design optimization: survey of recent developments. citeseer.ist.psu.edu/sobieszczanski-sobie96 multidisciplinary.html

Spath D, Neithardt W, Bangert C (2002) Optimized design with topology and shape optimization. Proceedings of the Institution of Mechanical Engineers, part B. Eng Manuf 216(8):1187

Woon SY, Querin OM, Steven GP (2000) Application of the fixed-grid FEA method to step-wise ga shape optimisation. In: Engineering design optimization. Proceedings of the second ASMO–UK /ISSMO conference, Swansea, UK, pp 265–272