ANNALS OF THE ORADEA UNIVERSITY
Fascicle of Management and Technological Engineering
ISSUE #1, MAY 2015, http://www.imtuoradea.ro/auo.fmte/

SHAPE AND SIZE OPTIMIZATION OF A GEARBOX BRACKET USING GENETIC ALGORITHMS

Razvan CAZACU[1], Lucian GRAMA[2], Mihai Dan GROZA[3]
[1] Petru Maior University of Targu Mures, Industrial Engineering and Management Department, razvan.cazacu@ing.upm.ro
[2] Petru Maior University of Targu Mures, Industrial Engineering and Management Department, lucian.grama@ing.upm.ro
[3] University of Oradea, Industrial Engineering Department, grozamd2002@yahoo.com

Abstract**—**This paper represents a case study regarding the structural optimization of a typical gear bracket used in automobiles. The functional specifications mimic those found in real gear brackets, both in terms of geometry and boundary conditions, including the loading, thus a direct comparison with such existing brackets can be made. The optimization procedure uses an original implementation of genetic algorithms, as found in the Object Oriented Genetic Algorithm framework for MATLAB. For the evaluation of solutions, the FEA software Abaqus was used, in conjunction with the scripting language Python. The results show the methodology can lead to lighter parts, while keeping the same strength and rigidity requirements. It also allows the automation of a task that would otherwise require important design efforts, without guaranteeing the same quality of solutions.

Keywords**—**shape optimization, gear bracket, genetic algorithm, Abaqus, MATLAB

I. INTRODUCTION

SHAPE and size optimization is usually carried out as the last step in structural optimization, dealing with fine-tuning geometries for which a general topology has already been established. In finding a final design for a given problem, topology optimization has an impact of up to 70% [1], thus being the main structural optimization drive force. However, results given by topology optimization usually need to be interpreted, requiring further processing, with notable but still insufficiently developed exceptions [2], while those given by shape and size optimization are most of the times final CAD designs [1]. More, as shown in [3], most topology optimization algorithms, including the most popular one in the industry: Solid Isotropic Material with Penalization (SIMP) [4], lead to globally feasible designs, but which violate the stress requirements at certain points where important stress concentrations appear. That is to be expected, as these methods optimize the total strain energy under a fixed volume fraction, which is a measure of rigidity rather than stress. In these conditions, the step of shape and size optimization is absolutely necessary as the final procedure in functional geometry optimization, in order to achieve truly optimized and feasible solutions. While this can be achieved both by manual iterations [2] or by any other optimization techniques, genetic algorithms (GA) have been proven to be a very effective shape optimization method [5-10]. There are numerous GA tools available for all programming languages and platforms. The problem of shape and size optimization implies using such a tool in conjunction with a FEA software for the evaluation and comparison of optimality of proposed solutions [8], [9], [11].

This research uses the original Object Oriented Genetic Algorithm (OOGA) for running the GA, an object-oriented programming (OOP) platform proposed by the author for the implementation of GA in MATLAB [9]. The platform allows for the rapid and powerful set up of common GAs, offers many options to configure and customize their behaviour and allows for a seamless integration with the corresponding FEA software.

For the FEA evaluation of the objective functions, two common approaches are:

1) using the APDL parametric language of ANSYS [8];

2) using the Simulia Abaqus FEA software in relation with the Python scripting language for the variation and simulation the modeled parametric solutions [9], [11].

In this paper we chose the second approach, considering both the ease of use and effectiveness of Python scripts and the advanced modeling and simulation capabilities of Abaqus.

II. PROBLEM FORMULATION AND SET UP

The structure chosen for this study is a typical bracket used to support the gear box of auto vehicles. The technical specifications impose 3 supports and 1 connection point and the proposed topology is indicated in Fig 1. This topology resembles the one currently used by the auto manufacturer but proposes slight modifications in the number and position of surfaces and ribs. The steel screws at the 3 supports have a fixed diameter of 10 mm and are modeled as 1D beam elements with the ends connected to the parts respective surfaces and fixed at the bottom. The part is assumed to be made of the aluminum alloy ENAC-AlSi9Cu3(Fe), having the following material characteristics:

1) Material density: $\rho = 2.56\, t/m^3$;

2) Young modulus: $E = 0.7 \cdot 10^5\, N/mm^2$;

3) Conventional yield strength: $R_{p0.2} = 140\, N/mm^2$.

4) Tensile strength: $R_m = 240\, N/mm^2$.

The bracket is subjected to 2 load combinations:

1) LoadX – The load sustained by the bracket in case of brake or impact. Beside the horizontal load, the gravitational load (supported mass) is also present.

2) LoadZ – The vertical, gravitational load carried out by the bracket.

The technical regulations require the part to be verified for 3 load cases:

1) Normal – the maximum stress must not exceed half of the conventional yield strength ($\sigma_{max} \le 0.5 \cdot R_{p0.2}$ *).*

2) Incidental – the maximum allowed stress is the conventional yield strength ($\sigma_{max} \le R_{p0.2}$ *).*

3) Accidental – the maximum allowed stress is the tensile strength ($\sigma_{max} \le R_m$ *).*

For this study we assumed the incidental load case and considered the conventional yield strength ($R_{p0.2}$) as the limit stress. The actual loads in the 2 load combinations are given in TABLE I below.

TABLE I
LOAD COMBINATIONS

Load Combination	Fx	Fz
LoadX	-8250 N	-1200 N
LoadZ	0	-6000 N

The topology of the part, along with the boundary conditions and load directions are indicated in Fig. 1.

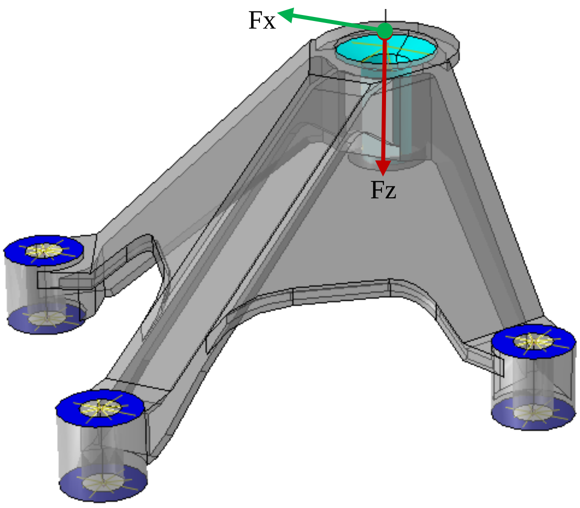

Fig. 1. Model topology and boundary conditions.

Based on the imposed topology, a set of parameters is attached to the geometry. We considered 17 parameters to describe the essential parts of the structure. These are generally edge lengths, arcs radii, points positions, plates and ribs thicknesses. The parameters, along with their imposed bounds (limits for their values) and the discrete increments in their domain, are listed in TABLE II.

TABLE II
PARAMETERS DESCRIPTION AND BOUNDS

Parameter Name	Parameter Description	Bounds	Inc
HConnPlate	height of upper plate	2 - 5	0.5
WConnPlate	width of skew upper plate	2.5 - 3.5	0.1
WCenterSkewPlate	width of ceneter skew plate	1 – 4	0.1
WCenterSuppPlate	width of center support upper plate	3 – 5	0.1
WLRSkewPlates	width of left and right skew plates	1 - 4	0.1
WLRSuppPlates	width of left and right support upper plates	12 – 16	0.1
SkewLRDCenter	center distance at side skew plates	10 – 30	1
SkewLRDSide	side distance at side skew plates	20 – 60	1
SkewLRShapeF	shape factor at side skew plates	0 – 1	.01
DTopBackPlate	center disctance at back plates	20 – 40	1
WBackPlates	width of back plates	3 – 7	0.1
HFrontPlates	height of front plates	6 – 8	0.2
WCenterLRPlates	width of center side plates	1 – 4	0.1
HTopCenterLRExtra	vertical distance for extra ribs	20 – 35	1
DTopCenterLRExtra	center distance for extra ribs	10 – 20	1
DInterCenterLRExtra	intermediate distance for extra ribs	10 – 30	1
HInterCenterLRExtra	intermediate height for extra ribs	6 - 20	1

The GA routine was applied using the MATLAB framework developed by the author in a previous work: Object Oriented Genetic Algorithm (OOGA) [9]. The connection with the Abaqus software used to evaluate the candidate solutions with FEA was possible due to the Abaqus connection class built in OOGA and with the help of a Python script responsible for the model geometric variation, remeshing, analysis and postprocessing.

The objective function of the optimization is the minimization of the total volume of the structure (as a measure of its mass), under the constraint of maximum allowed stress (conventional yield strength). The constraint was handled by the means of the penalty function developed in [7].

Because the genetic algorithms are stochastic in nature, the end results are unpredictable and usually unrepeatable. To account for this fact and verify that the optimum solution is really the global optimum and not just a local minimum, we have run the algorithm 4 times (called Run A through Run D), varying some of the GA from one run to the other, as shown in TABLE III. These parameters are the total number of generations, the number of individuals in the population (population size), the mutation method and the mutation parameter final value.

A comprehensive list of classic mutation operators is given in [12]. We chose for this study the Uniform mutation for Run A and Run B, and Polynomial mutation for Run C and Run D. The actual implementation of these mutation operators in OOGA, detailed in [9], is slightly different than their original form, in order to enhance their flexibility and versatility. OOGA also allows the variation of the parameter value over generations, in order to keep the exploration capability of mutation in earlier generations but improve its exploitation capabilities in later generations.

TABLE III
GA SETTINGS FOR RUNS A-D

Setting	Run A	Run B	Run C	Run D
Number of generations	51	101	101	81
Population size	42	32	40	40
Mutation method	Unif.	Unif.	Poly.	Poly.
Mutation parameter final value	0.1	0.1	8	1

For all runs, the common GA settings used are given in the list below:

1) Crossover method: 3-point (generalization of single-point and 2-point crossover)
2) Crossover probability: 80% of the new individuals undergo crossover;
3) Elitism: 2 individuals are automatically passed to the new generation, ensuring the best solutions survive;
4) Individual mutation probability: 100% (all individuals have a chance of mutating);
5) Gene mutation probability: 20% of the genes of each individual mutate;
6) Mutation parameter initial value: 1 (affecting the mutation probability; the final value for each run is given in TABLE III);
7) Selection: stochastic uniform (combination between deterministic and stochastic);
8) Fitness scaling: rank based (each individual receives a scaled fitness score indirectly proportional to the square root of its rank).

III. RESULTS

As shown in the previous section, the optimization procedure was run 4 times, with different settings. The final best solution for each case is described in TABLE IV. This shows the final values of the 17 parameters, the maximum equivalent von Mises stress, the volume of the part and its mass. As can be observed, all the runs lead to similar models, most of the parameters being the same or almost the same across runs and the final mass of the part being practically the same. At the same time, the maximum von Mises stress in the structure is close to the limit $\sigma_{lim} = 140 MPa$.

TABLE IV
OPTIMUM SOLUTIONS

Parameter Name	Run A	Run B	Run C	Run D
HConnPlate	3	3	3	3
WConnPlate	2.6	2.9	2.5	2.5
WCenterSkewPlate	2.5	2.5	2.5	2.5
WCenterSuppPlate	4.8	4	4	3
WLRSkewPlates	2	2	2	2
WLRSuppPlates	14.5	14.4	14.8	16.0
SkewLRDCenter	25	25	25	20
SkewLRDSide	45	45	45	44
SkewLRShapeF	0.86	1	1	1
DTopBackPlate	33	33	33	33
WBackPlates	4	4	4	4
HFrontPlates	6	6.2	6	6
WCenterLRPlates	2	2	2	2
HTopCenterLRExtra	25	25	26	25
DTopCenterLRExtra	10	13	11	10
DInterCenterLRExtra	13	11	10	10
HInterCenterLRExtra	10	10	10	10
MaxMises (MPa)	139.6	139.1	139.3	139.1
Volume (10^3 mm^3)	123.7	123.8	123.3	123.4
Mass (kg)	0.309	0.309	**0.308**	0.308

The similarity between end results shows the GA optimization procedure is reliable. We chose to detail the results only for Run C, as this gives the best end result, even if by a very small margin.

In order to illustrate the general convergence of the solution, Fig. 2 depicts the evolution of the best individual's score over the 101 generations performed in Run C. The evolution is typical for a GA, with a more pronounced convergence in the early stages, when the domain is thoroughly explored and a slower convergence towards the end, when the main interest is exploiting and fine tuning the best solutions.

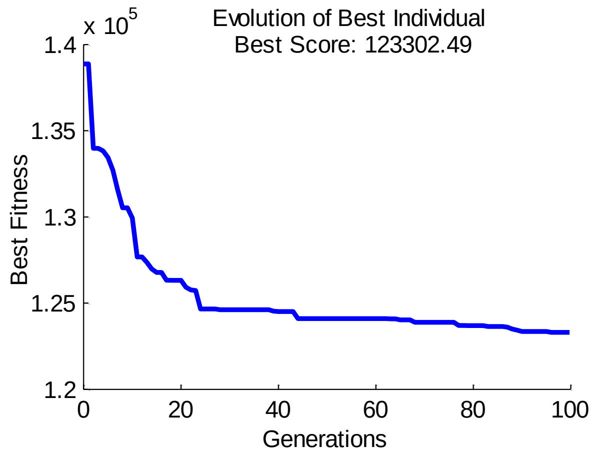

Fig. 2. Evolution of best individual over generations (Run C).

It is worth noting that the end mass is about 0.308 kg, at a volume of about 123.5 x 10^3 mm^3. The real gearbox bracket this model is based upon has a mass of a little above 0.5 kg, which is considerable more than the solution found in this research. Part of this difference can be explained by the slight differences in initial specifications and by the fact this model is only in functional stage, while the real gearbox is the final product, obtained after experiencing technological redesigns. However, these factors can’t account for the whole difference, the mass reduction aimed by the study being thus achieved. However, the exact magnitude of this reduction is hard to be assessed exactly in the absence of detailed and precise data about the real model.

The distribution of the equivalent stresses in the optimum model is depicted in Fig. 3. A view from the bottom of the part was chosen, as that is where the highest stresses appear. Considering there are some areas with stress concentrations and some with low stress, a new design with a different topology could possibly offer opportunities for even lighter models.

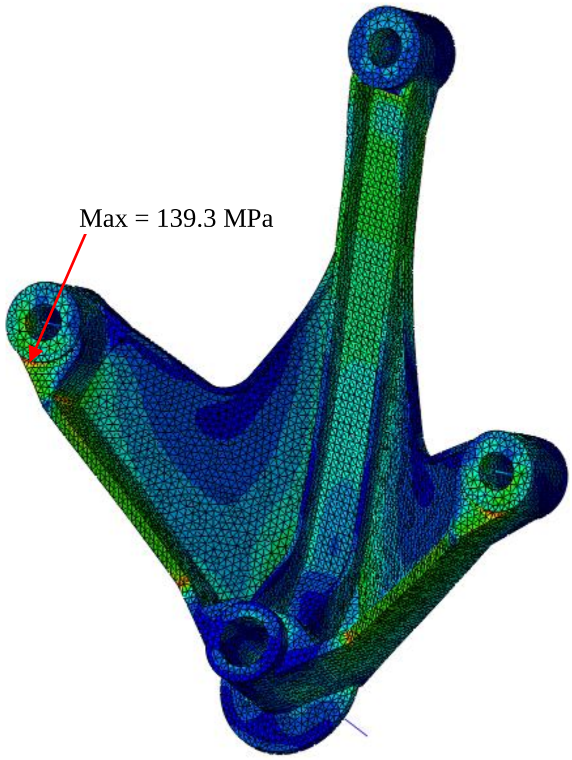

Fig. 3. Equivalent vonMises stress distribution for the LoadZ load case of the optimum model (Run C).

IV. CONCLUSIONS

The results obtained in this research show the procedure of functional structural optimization using genetic algorithms can lead to improved designs of auto parts. Besides being able to output lighter structures in the same strength and rigidity conditions, the algorithms allow the automation of this process that would otherwise take much more time and would not guarantee true optimal solutions.

Considering all the software needed to run the optimization technique should be part of the endowment of any design department, the only additions needed are the tools presented in [9-10]. As shown here, these can successfully be applied to real-world design problems in the automotive or any other industry.

REFERENCES

[1] L. Harzheim, G. Graf, "A review of optimization of cast parts using topology optimization - Part 1", *Structural and Multidisciplinary Optimization*, vol. 30, no. 6, 2005, pp. 491-497.
[2] D. Weiss, "Feature-based spline optimization in CAD", *Structural and Multidisciplinary Optimization*, vol. 42, no. 4, 2010, pp. 619-631.
[3] G. Dima, I. Balcu, M. Zamfir, "Method for Lightweight Optimization for Aerospace Milled Parts – Case Study for a Helicopter Pilot Lightweight Crashworthy Seat Side Struts", *Procedia Techology*, vol. 19, no. 1, 2015, pp. 161–168.
[4] G.I.N. Rozvany, "A critical review of established methods of structural topology optimization", *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, 2009, pp. 217-237.
[5] S. Sanchez-Caballero, M.A. Selles, R. Pla-Ferrando, A.V. Martinez Sanz, M.A. Peydro, "Recent Advances in Structural Optimization", *Annals of the Oradea University, Fascicle MTE*, vol. 11, no. 1, 2012, pp. 2.118-2.127.
[6] R. Cazacu, L. Grama, "Comparative study for the size optimization of a tapered pipe cantilever beam using genetic algorithms", *Proceedings of the Annual Session of Scientific Papers ”IMT Oradea”*, vol. 13(23), no. 1, 2014, pp. 81-84.
[7] R. Cazacu, L. Grama, "Steel truss optimization using genetic algorithms and FEA", *Procedia Technology*, vol. 12, 2013, pp. 339–346.
[8] J.E. Rodriguez, A.L. Medaglia, J.P. Casas, "Approximation to the optimum design of a motorcycle frame using finite element analysis and evolutionary algorithms", *IEEE Systems and Information Engineering Design Symposium*, 29 April, 2005, pp. 277-285.
[9] R. Cazacu, L. Grama, I. Mocian, "An OOP MATLAB Extensible Framework for the Implementation of Genetic Algorithms. Part I: The Framework", *Procedia Technology*, vol. 19, no. 1, 2015, pp. 193-200.
[10] R. Cazacu, L. Grama, I. Mocian, "An OOP MATLAB Extensible Framework for the Implementation of Genetic Algorithms. Part II: Case Study", *Procedia Technology*, vol. 19, no. 1, 2015, pp. 201-206.
[11] L. Luo, M. Zhao, "The Secondary Development of ABAQUS by using Python and the Application of the Advanced GA", *Physics Procedia*, vol. 22, 2011, pp. 68-73.
[12] P.H. Tang, M.H. Tseng, "Adaptive directed mutation for real-coded genetic algorithms", *Applied Soft Computing*, vol. 13, no. 1, 2013, pp. 600-614.