

Comparison of Strategies for the Optimization/Innovation of Crankshaft Balance

Albert Albers¹, Noel Leon², Humberto Aguayo² and Thomas Maier¹

1 IPEK – Institute of Product Development Karlsruhe, University of Karlsruhe, Germany

2 CIDT, – Center for Innovation in Design & Technology, Tecnológico de Monterrey, Mexico

Abstract. Engine crankshafts are required to be balanced. The balance of a crankshaft is one of several parameters to be analyzed during the design of an engine, but certainly a poor balance leads to a low life time of the whole system. It is possible to optimize the balance of a crankshaft using CAD and CAE software, thanks to the new optimization tools based on Genetic Algorithms (GA) and tools for the integration of the CAD-CAE software. GAs have been used in various applications, one of which is the optimization of geometric shapes, a relatively recent area with high research potential. This paper describes a general strategy to optimize the balance of a crankshaft. A comparison is made among different tools used for the sustaining of this strategy. This paper is an extension of a previous paper by the authors [1] but now different tools are being included to improve the performance of the strategy. The analyzed crankshaft is modeled in commercial 3D parametric software. A Java interface included in the CAD software is used for evaluating the fitness function (the balance). Two GAs from different sources and platforms are used and then they are compared and discussed.

1 Introduction

This paper compares different optimization and integration tools that can be used to build up a design strategy on crankshaft design and optimization. Though advanced CAD software has its own optimization engines, these can not be compared to more powerful genetic algorithms, such as those inside DAKOTA (Design Analysis Kit for Optimization Applications) [2] developed at Sandia Laboratories or even VGGA (Virtual Gene Genetic Algorithm)[3] developed at Tecnológico de Monterrey. It is therefore necessary to develop a proper interface to link the GAs to the CAD models. This work compares various approaches to this task, from commercial software to a JAVA interface required to integrate the genetic

algorithms with the CAD model of the crankshafts. A comparison of recently obtained results is presented in this paper.

1.1 Known works

The splinization approach to optimizing designs using GAs is relatively new; some examples are the computer design and optimization of cam shapes for diesel engines [4]. In this case the objective of the cam design was to minimize the vibrations of the system and to make smooth changes to the splined profile. Another example is the work developed at the Institute of Product Development (IPEK) at the University of Karlsruhe on the search for an assembly array of several tubular components whose trajectory is described by splines.

1.2 Our work

This paper continues from a previous work: “Computer Aided Innovation of Crankshafts using Genetic Algorithms” [1], where a method for determining the design imbalance of CAD modeled crankshafts inside CAD software is described. In this previous work a procedure for a balanced design strategy was introduced. The mass properties required to calculate the balance (m_g , r_{gy} , r_{gz} and the inertia products I_{xy} and I_{xz}) of the crankshaft models can be obtained from parametric CAD software, which has special commands in its advanced modules, for calculating the balance as a response for fitness evaluation. If a target for imbalance is specified the difference between the target and the current imbalance is a fitness function that has to be minimized by making modifications to the crankshaft geometry. In order to make geometry modifications it was decided to substitute the profile of the counterweights with splines from its original “arc-shaped” design, because cubic splines allow smooth shape changes via control points with continuous second derivatives, a desired property for material fluency during the manufacturing process. The x and y coordinates of the control points can be parametrically manipulated by an optimization algorithm, i.e. by a Genetic Algorithms (GA).

2 Preliminaries

2.1 Problem Description

The figure below shows a counterweight profile of a V6 engine crankshaft. From a previous sensitivity analysis it was found that defects in the profile (under filling) of the outermost counterweights, number 1 (CW1) and 9 (CW9), have the greatest influence on the balance of the crankshaft. It was decided to begin with codification of the “Y” coordinates of 4 control points, each upper and lower profile from both counterweights, resulting in 16 control parameters. The variation of these control parameters results in a balance response. The fitness function selected as the response is an equally weighted function of the differences between the specified

target for imbalance and the current imbalance on the two external sides of the crankshaft (CW1 and CW9).

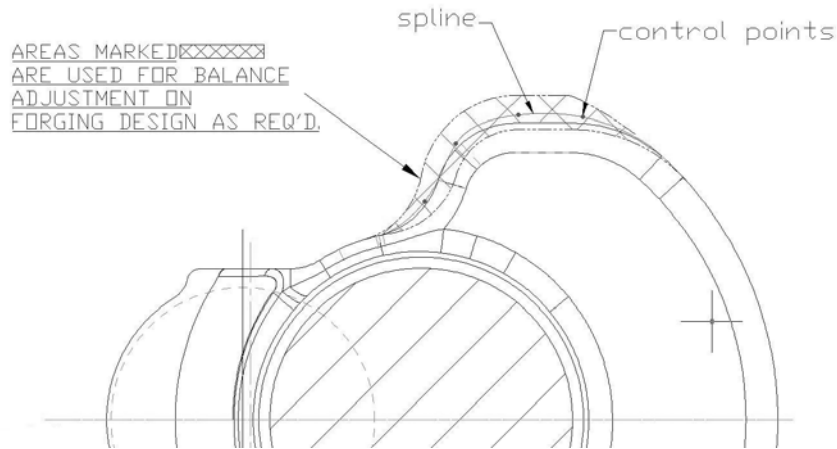


Fig. 1. Profile of a counterweight represented by a spline and its control points

This method is called “Goal Programming”, in which the designer has to assign targets or goals to be achieved for each objective function [5]. These values are incorporated into the problem as additional constraints. The algorithm will then try to minimize the absolute deviations from the targets to the objective functions. The simplest form of this method may be formulated as follows:

$$\min \sum_{i=1}^k |f_i(x) - T_i|, \text{ subject to } x \in X$$

where T_i denotes the target or goal set by the designer for the i th objective function $f_i(x)$ and X represents the feasible region. The criterion, then, is to minimize the sum of the absolute values of the differences between the target values and the actually achieved values of imbalance on the two external counterweights.

2.2 Strategy and Approach No. 1: CAD Model linked to VGGA via Commercial Integration Software

Using VGGA, a population of 50 individuals is generated randomly. Each individual is a gene containing a set of 16 parameters, codified in lineal mapping 0s being the lower limits and 1s the upper limits. Once generated, the phenotypes (de-codified) are taken by the commercial software, parsed to the macro file read by the CAD software and run automatically to generate a fitness result; this fitness result is read and sent back to VGGA, where it is assigned to the corresponding individual.

After the whole generation has been evaluated, tournament selection plus other genetic operators produce the next generation. Figure 2 shows the proposed sequence of steps.

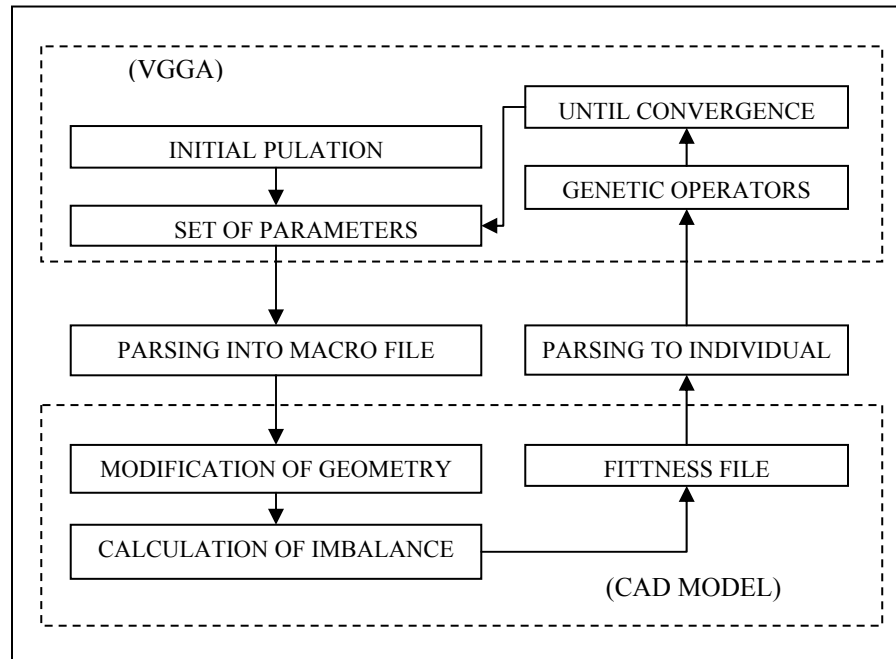


Fig. 2. Integration of VGGA with the CAD Model using commercial software

The Genetic Algorithm performed well during the initial runs, as it was capable of delivering a geometry that improved the fitness function in each generation. However, because commercial software was used for the integration, there was low flexibility and the CAD software started up on each individual evaluation, causing a long standing time, which made large runs prohibitive. Two additional approaches for the same strategy are presented, and a comparison has been made.

3 Development of Alternative Approaches

3.1 Strategy and Approach No. 2: CAD Model linked to VGGA via JAVA

The parametric CAD software used in the project comes with an interface programmed in JAVA, called J-Link, and some of the subroutines were adapted to substitute the commercial software from the first approach, in a sequence similar to that shown in figure 2 but with some differences:

1. The control of the Genetic Algorithm is still made by VGGA, but the evaluation of the fitness function in the CAD software is controlled by JAVA, instead of the commercial integration software.
2. The CAD software is kept running in parallel to VGGA. The CAD software therefore does not need to stop its execution and start up every time the fitness function is evaluated. Once the fitness function file is ready, VGGA reads it and assigns it to the corresponding individual to continue with the algorithm.

The main advantage of this approach was that the need to start up the CAD software after each evaluation is eliminated, and it shut downs and starts again automatically only when the set of parameters causes a geometry that cannot be regenerated (for example, a CAD model that conflicts with characteristics form areas other than counterweights). When this happens, the fitness function is assigned a value of zero so that the individual does not to continue to the next generation. Some runs were made with a similar set of parameters as with the commercial software. Being able to run a substantial number of evaluations, VGGA showed a convergence to a fixed fitness value, as shown in figure 3.

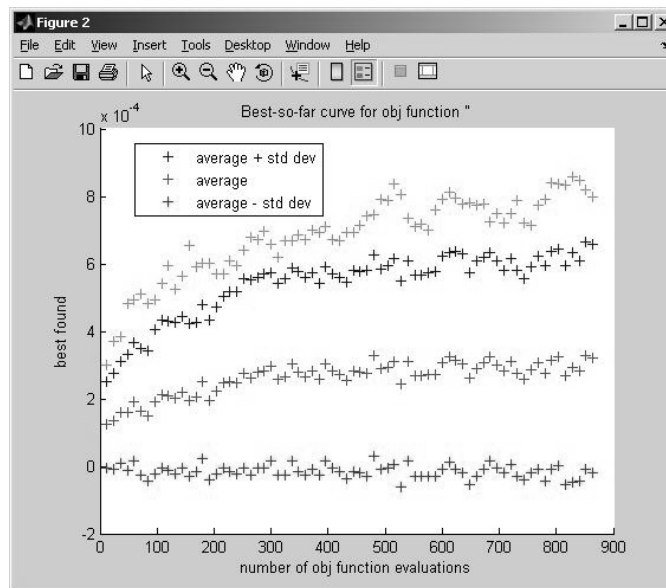


Fig. 3. Evolution of the fitness function using VGGA

It should be noted that the graph does not show a continuous increase, which would have been expected as the balance reaches the target. Because the target balance could not be attained inside the design constrains defined for the geometry of the counterweights, some trials were performed allowing the spline to trespass the geometry constraints. However, although improved results were obtained, the convergence was not yet satisfactory. It should be noted that the shapes that trespass the boundaries, although not satisfactory from the forging point of view, tend to

“separate” the balancing mass out from the crankshaft center axis. This kind of solution has already been implemented in the design of other engine crankshafts. This behavior leads to the conclusion that a different algorithm was required for validating the convergence. It was therefore decided to substitute VGGA, and to continue to the next step of the study using a GA solver inside the DAKOTA toolkit.

3.2 Strategy and Approach No. 3: CAD Model linked to DAKOTA via JAVA

DAKOTA is a parameter solver developed at SANDIA Laboratories with the original goal of providing a common set of optimization tools for engineers who need to solve structural and design problems, including Genetic Algorithms. The proposed sequence of steps is shown in figure 4, which shows the parallel execution of DAKOTA and CAD Software via JAVA.

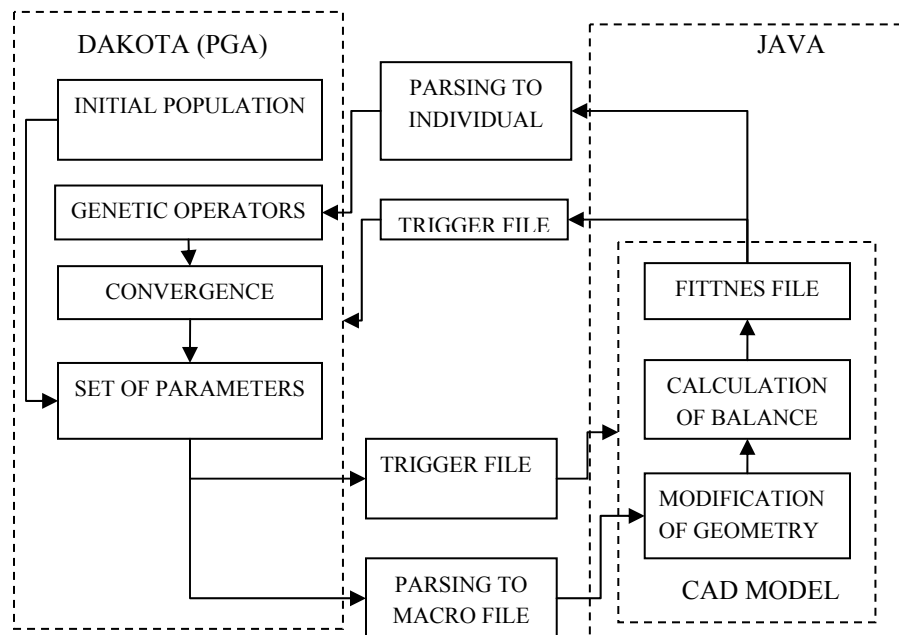


Fig. 4. Integration of DAKOTA with CAD Software using JAVA

In order to substitute VGGA by a DAKOTA GA it was necessary to make some adjustments to the strategy:

1. A DAKOTA GA was chosen that solves for minimization, instead of making a maximization, so the fitness function is the inverse of the result used in VGGA, aiming at a zero value (important when comparing the two evolution graphs).
2. Most of the parameters used in VGGA were kept (mutation and crossover ratios, etc) but now the solver (named `pga_real`) is from the DAKOTA

“Stochastic Global Optimization” library, a genetic algorithm for real numbers.

The evolution of the fitness function converged to a non-zero value, as can be seen in figure 5. The formatting of the graph is different than in VGGA, because different graph-making tools were used.

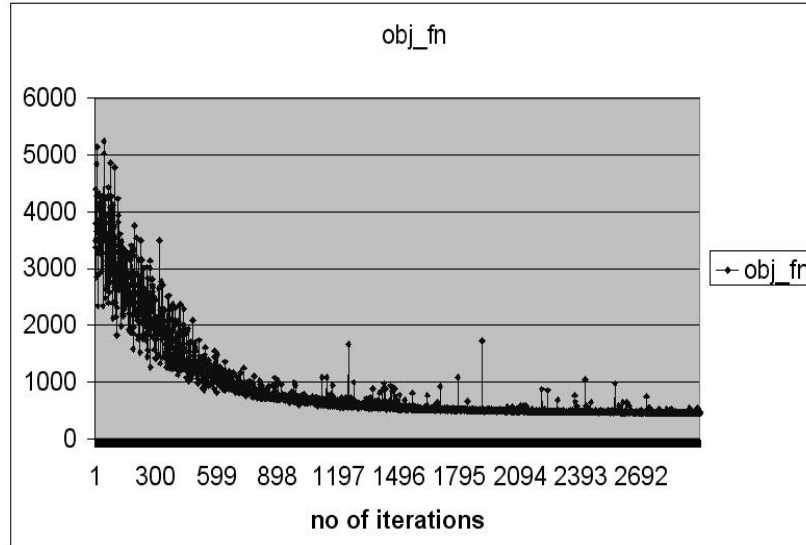


Fig. 5. Evolution of the fitness function using DAKOTA

It should be noted that it was not possible either with VGGA or with DAKOTA, to reach the balance target inside the design constrains of the parameters that control the splinized counterweight profiles. After analyzing the balancing behavior of some individuals during the evolutions of the two GAs, it was observed that when a good balance was reached in CW1, the balance in CW9 worsened, and vice versa. This conflicting behavior needed to be confirmed using the Pareto Frontier method.

The balances on both sides of the crankshaft (CW1 and CW9) were defined as two independent objective functions. Because no data on this condition had been collected on the previous runs, a new run was required. We chose a Multi Objective Genetic Algorithm (MOGA) from the DAKOTA toolkit to generate the required data. Figure 6 shows a graph with the value of the first function (imbalance of CW1) in the x-axis and the value of the second function (imbalance of CW9) in the y-axis. This graph is known as the Pareto Frontier

There is a generalized notion in multi-objective optimization that no “optimal” solution can be attained but that there is a set of optimal solutions lying on a line that prevent the functions reaching the “ideal” at the same time. Here, different theories on “trade off” management arise. Further discussion can be conducted on how innovation theories can “jump over” the technical contradiction and how Genetic Algorithms could help in this issue. From the point of view of a classical genetic algorithm the closest approach dealing with the “ideal” concept is related to an

“Alternative Ideal”: to find the “ideal point”, a decision must be taken by exploring the limits along each constraint so that an alternative ideal can be defined [5].

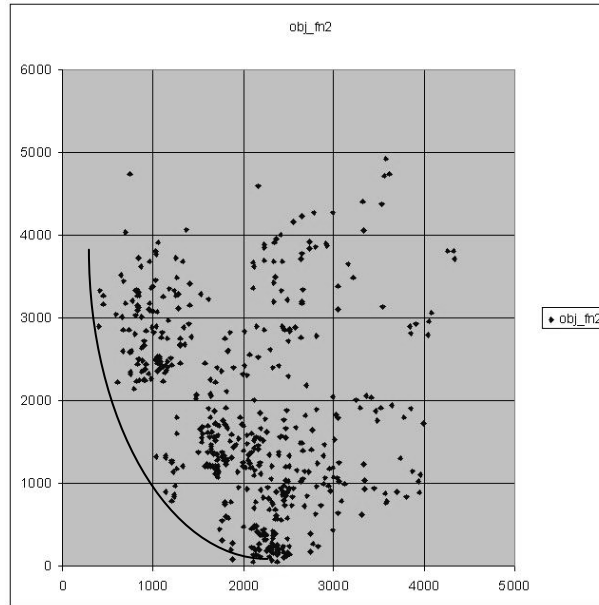


Fig. 6. Pareto Frontier between conflicting functions

In order to have a visual feeling about the way the algorithm is performing, an individual from the most centered value on the Pareto Line of the graph was selected. The resulting geometry of that individual is shown in figure 7. In these pictures the profiles of the counterweights come close to the limits imposed by the original design (sketched in blue). It is inferred that, in order to reach the balance target, it may be necessary to reconsider the geometric constraints. As can be seen in this case, sharp edges arise in the profile, and this is not good for the forging of the crankshaft. New constraints to the optimization strategy have to be formulated, taking control of the curvature of the splines in such a way that no sharp edge develop and forging process is not worsened. This is only a single step forward in a field of specific functionality, but expanding the solution to open design dimensions [6], there will be a need for factors that exploit the parametric geometry of the CAD system with innovative concepts.

The proposed system has two objective functions (imbalance-to-target on CW1 and imbalance-to-target on CW9), and the original geometric constraints are widened (lower and upper limits of the control points of the splines, plus constraints on the curvature of the splines).

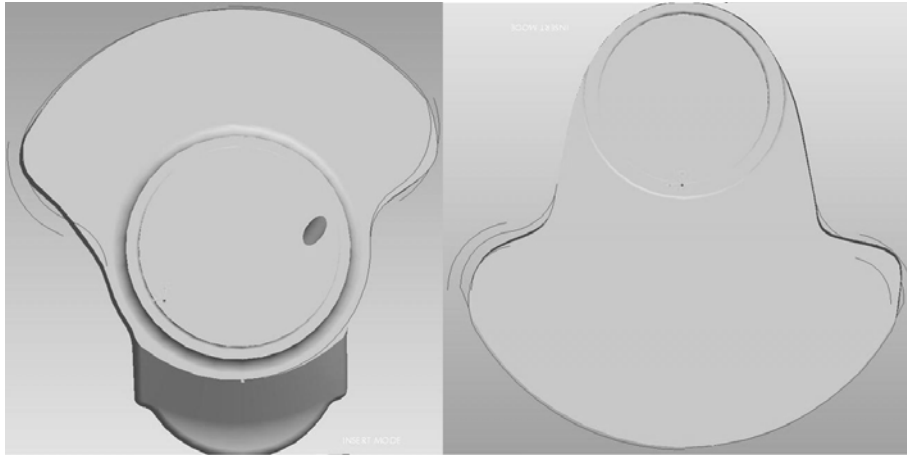


Fig. 7. Profile Shapes of CW1 and CW9 from an individual in the Pareto Frontier

3 Conclusions and further research

Both VGGA and DAKOTA converged in about the same number of evaluations. The use of the JAVA interface prevented the need to start up the CAD software during each evaluation that significantly improved the run time. MOGA from DAKOTA allowed developing the development of a Pareto frontier from which important conclusions were extracted:

- It is necessary to prevent the sharp edges of the splines by adding an extra restriction that controls the curvature of the shape. This can be done by adding an evaluation during each individual run, in an additional solver.
- Simulation of the forging process is a required next step in order to define a relationship between good shapes-curvature and manufacturability. This becomes significantly important when a proposed design outside the initial shape restrictions needs to be justified in order not to affect formability.
- Stress analysis of the crankshaft is a preferred approach to be added either a restriction or an objective during modification-evolution of the geometry of the crankshaft in order not to affect its functionality.

This paper defines the basis and the beginning of a strategy for developing crankshafts that will include the manufacturability and functionality to compile a whole Multi-objective System Optimization. The use of the Java interface will allow the control and manipulation of additional software to be required during restriction evaluations. A medium-term objective is to implement innovation concepts, i.e. operators, into the evolutionary mechanisms of computer algorithms in order to trespass the barriers imposed by conflicting objectives (Pareto frontier).

Acknowledgements

The authors acknowledge the support received from Tecnológico de Monterrey through Grant number CAT043 to carry out the research reported in this paper.

References

1. Aguayo, Humberto, and Noel Leon. 2006. Computer Aided Innovation of Crankshafts Using Genetic Algorithms. Chap. in *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*. IFIP International Federation for Information Processing. 471-476. Springer Boston.
2. Eldred, Michael, Anthony Guinta, William Hart, John Eddy, and Josh Griffin, Technical Report. October 2006. *DAKOTA Version 4.0 User's Manual*. Albuquerque, NM, USA: Sandia National Laboratories.
3. Valenzuela-Rendón, Manuel. The Virtual Gene Genetic Algorithm. Chap. in *Genetic and Evolutionary Computation GECCO 2003*.
4. Lampien, J. JULY 2003. Cam shape optimisation by genetic algorithm. *Computer-Aided Design* 35, 8: 727-737.
5. Coello Coello, Carlos Artemio. 1996. An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design. Doctor of Philosophy diss., Tulane University.
6. Cueva de Leon, Jose Maria. 2006. Automatic shape variations for optimization purposes. Master in Sciences thesis, Tecnológico de Monterrey (ITESM).