INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
MONTERREY CAMPUS

ENGINEERING DIVISION
ENGINEERING GRADUATE PROGRAM

TECNOLÓGICO DE MONTERREY®

RESEARCH DISSERTATION:

# Integration of structural shape modeling, optimization and analysis methods: foundations of a new computer aided inventing strategy and application to an engine crankshaft.

PRESENTED AS PARTIAL FULFILLMENT TO OBTAIN THE DEGREE OF:
DOCTOR IN ENGINEERING

PRESENTED BY:

HUMBERTO AGUAYO TELLEZ

MONTERREY, N.L. DECEMBER 2009

Research dissertation

Humberto Aguayo

# Integration of structural shape modeling, optimization and analysis methods: foundations of a new computer aided inventing strategy and application to an engine crankshaft.

Advisor: Dr.-Ing. Noel Leon

Printed in: Monterrey, Mexico

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY**
**CAMPUS MONTERREY**

**ENGINEERING DIVISION**
**ENGINEERING GRADUATE PROGRAM**

Committee members recommend that the research project presented by Humberto Aguayo Téllez be accepted as a partial requirement to obtain the academic degree:

**Doctor in Engineering**

APROVED BY:

Dr. Noel León Rovira
Assessor

Dr. Manuel Valenzuela Rendón
Committee member

Dr. Alberto Hernández Luna
Committee member

Dr. Jorge Armando Cortés Ramírez
Committee member

Dr. Víctor Hiram Vázquez Lasso
Committee member

Dr. Alex Elías Zúñiga
Director of Doctor in Engineering Program.

**December, 2009.**

# Abstract

Structural optimization has achieved a level of development that makes it possible to go beyond the current limits for optimizing shapes of parts. This kind of optimization also allows designers to improve the traditional design process, helping them to make decisions regarding new inventive shapes and solutions that arise during the shape variation and simulation process. Designers can now take advantage of not only using parametric CAD models to analyze the influence of geometric parameters variations, but also the effect of shape variations on the required performance of their designs. This thesis presents a strategy that allows the designer to perform an automatic search process for crankshaft design objectives based on genetic algorithms, integrated with the simulation of crankshaft behavior, with respect to its balancing and torsion vibrations. At the same time, it looks for an optimal forging process, improved with respect to the design methods used until now. An important characteristic of the strategy presented in this thesis is the automation of the search process, as genetic algorithms through crossbreeding and mutations of the counterweight profile, control the variation of shapes in search for the target balance. At the same time, seek to achieve a target position of the crankshaft's center of gravity and achieving an optimally forgeable shape.

Keywords: Genetic algorithms, Splines, Crankshaft, TRIZ, FEM

# Acknowledgements

I thank to all the people who have contributed to this achievement. I specially thank to my parents Lourdes and Humberto for their love and support. I want to express my gratitude to my advisor Dr Noel Leon for encouraging me to study the Doctorate. I also thank all the professors from Campus Monterrey, Dr. Manuel Valenzuela Rendón, Dr. Alberto Hernández Luna, Dr. Jorge Armando Cortés Ramírez and Dr. Víctor Hiram Vázquez Lasso, members of my committee. I thank Prof. Dr. Ing. Albert Albers for his support during my stay at Karlsruhe University.

# Dedication

To my beloved wife Karool, I could not have done it without you.

# Table of Contents

## List of Figures

# List of Tables

# Chapter 1 Introduction

The development of products using simulation with *computer-aided design and engineering*, contributes to the enhancement of an enterprise's inventive abilities. Even when the newest modeling systems are capable of representing complex physical situations, the optimization algorithms play an important role in the generation of significant improvements. Recently, computers have taken over large parts of the simulation in all fields where prediction of the performance is necessary. As explained by Dawkins [1], simulation is used when there is a difficult decision to make involving unknown quantities in the future.

This dissertation deals with the development of a strategy for the optimization and invention of products using *evolutionary design* strategies, CAD representation, CAE simulation and automated integration of all these elements. A significant part of this research work starts with the parametric shape optimization of products and then continues with the application of elements related to the concept of computer-aided innovation. In this introduction, an explanation is given of the context in which the tasks to optimize designs are formulated and the foundations of the strategy are established. There is a description of how current design optimization techniques are used and how they can be further developed by making use of geometric splines, in combination with *genetic algorithms*, to achieve a strategy that allows computers to modify the CAD model automatically and provide inventive results. The case study selected to carry out the proposed strategy is an engine crankshaft, manufactured in a closed forge process.

Crankshafts are used as case study because: they are one of the most difficult mechanical elements to design due to the complex dynamic relationships among its features. Crankshafts have a relevant impact in the performance of internal combustion engines and its manufacturing is a challenging task. The heart of a crankshaft geometry traditionally consisted on arc-shaped counterweight profiles. These counterweights have to balance the dynamic loads of the whole piston-rod-crank assembly. Additionally the forging process demands that the shape of the counterweights are designed for a better flow of the metal into die cavity and robustness in the balance and dynamic behavior of the crankshaft. The crankshaft is an element with different functionalities (transform movement into rotation, manufacturing) and physical behaviors (natural frequencies, imbalance); as all these characteristics conflict among each other, it is especially suitable to be the case study.

During the development of the dissertation, different software tools were used in order to execute the different tasks involved: CAD modeling, genetic algorithms, CAE simulation and programming in Java. The intended strategy is based on the simulation of different characteristics of a product. Automation of the process is also pursued, through the integration of the different software, in order to evaluate the emerging design alternatives. The aim of the strategy is to find a new way of performing the design process that can reduce the product development cycle and achieve superior performance compared to traditional trial and error strategies.

The next section is a sequential introduction of the methods and techniques deployed during the optimization/invention strategy for the development of the engine crankshaft. First, there is an explanation of evolutionary design, which is a defined group of techniques with similarities that can be seen as the starting point for the construction of the proposed method. The general kinds of mechanical optimization, from which the shape optimization approach is selected to be employed in conjunction with geometric splines, are also presented. Next, the state-of-the-art for engine crankshafts design via the analysis of patents is given. At the end of the section, the hypothesis and the objectives of the research in both the development of the strategy, and the solutions for the crankshaft per se are presented.

## 1.1 Evolutionary Design

A relatively new area of development called *evolutionary design* [2] has become an object of intensive research. Peter Bentley describes evolutionary design as a process capable of generating designs by changing topologies and shapes. In this way, an intricate design can arise through a slow, gradual improvement process.

Evolutionary design has its roots in computer science, design and evolutionary biology. It extends and combines CAD and analysis software, and borrows ideas from natural evolution. Evolutionary Computation to optimize existing designs (i.e. perform detailed design or parametric design) was the first type of evolutionary design to be tackled [3]. Although the exact approach used by developers of such systems varies, typically practitioners of evolutionary optimization usually begin the process with an existing design, and parameterize those parts of the design they feel need improvement.

To give a general context of earlier related approaches, a description of the different categories of evolutionary design that have been classified by Bentley: evolutionary optimization, creative evolutionary design and conceptual evolutionary design, will be helpful. Evolutionary optimization places great emphasis on finding solutions as close to the global optimal as possible [4]. Creative evolutionary design is concerned with the generation of novel designs [5]. In conceptual evolutionary design the relationships and arrangements of high-level design concepts are evolved by trying to generate novel preliminary designs. At the conceptual design stage, designers want to know if the lines of thought they follow are worth pursuing. This understanding can be in the form of underlying trends, existing solutions, trade-offs, etc. that have been addressed by different approaches [6]. Generative (or conceptual) evolutionary designs using computers to generate the designs form and not only a collection of predefined high-level concepts have the advantage of giving greater freedom to the computer. Typically, such systems are free to evolve any form capable of being represented, and the evolution of such forms may well result in the emergence of implicit design concepts. The concept of *emergence* related to conceptual design is related to the perceptual processes of interpretation and representation that can evolve into a novel design [7].

Evolutionary design mimics the way nature behaves by using evolutionary algorithms that change the forms and topologies of the design object. The biological analogy of such computational methods is based on some elemental rules of the evolution of species. Such rules of evolution state three vital factors for the existence of an evolutionary process:

- A reproductive process or replication ability.
- A selection process based on performance under given life conditions.
- Capacity of incurring on reproductive errors (replication with variations).

These factors can be used to simulate the biological evolution on a computer. In doing so, the solution set is represented by data in the computer. The algorithms that manipulate data structures use the biological analogy for selection, mutation and recombination. Although the term "simulated evolution" deliberately suggests an analogy with biological evolution, it is understood that the real biological processes are far more complex than any computer simulation, the simulated evolution is only an idealization of certain aspects of a biological system. Biological systems serve as an inspiration, but computers are not able to implement all biological phenomena, that even today has not been completely understood, such as the existence of parasites, which also have served as inspiration for optimization problems [8].

One of the components of evolutionary computation, the genetic algorithm, has shown a great potential for working on several real-world problems to the point of optimization; in reality, however, it is still quite far from realizing a system of matching the human performance, especially in creative applications such as architecture, art, music, and design [9]. The optimization of existing designs is relatively common, with the creation of artistic images and artificial life growing rapidly. Nevertheless, the development of evolutionary design tools is still at an early stage. So far, many genetic algorithms have been used and tested only in design problem solution with small scope. The research and development of design support tools using evolutionary computing technology are still ongoing and have potential for the development of new design technology.

The use of evolutionary computation to generate designs has taken place in many fields since the late 1980s. Designers have optimized selected parts of their designs using evolutionary computation. John Holland presented the pioneering formulation of genetic algorithms and described how the evolutionary process in nature can be applied to artificial systems using the genetic algorithm operating on fixed length character strings in *Adaptation in Natural and Artificial Systems* [10]. In this work, Holland demonstrated that a wide variety of different problems in adaptive systems (including problems from economics, game theory, pattern recognition, optimization, and artificial intelligence) are susceptible to reformation in genetic terms so that they can potentially be solved by parallel computation.

The use of genetic algorithms combined with 3D CAD packages and CAE simulation tools has proven to be useful for searching the design space for better solutions, but implementing it consumes a great amount of time and computational resources [11]. Current CAD/CAE packages are originally conceived as to mainly facilitate parametric variations of the parts. In cases where shape and/or topological changes of parts or assemblies are required, such changes have to be manually performed. Activities in the design process that are still performed manually instead of with the use of computers include preliminary design [12], the planning of assembly sequences of mechanical parts [13], and others.

## 1.2 Optimization of Mechanical Systems

The evolution of product development tools has been characterized by various trends, and the analysis of these trends offers useful hints for predicting next generation systems. In mechanical design, optimization tasks are used for structural optimization, which deals with the development of mechanical structures [14]. Optimization of continuous mechanical structures is much employed in industry, for shape and topology and a combination of the two [15]. The optimization of products and processes has been studied since the spread of computers as an aid for seeking “optimal” forms and shapes of product geometry. For example, when minimizing the weight of a mechanical component, restrictions have to be included to guarantee the stability of the structure (ex. stresses or natural frequencies). The objectives of structural optimization are as follows: (a) minimizing stress or weight; maximizing lifespan, stiffness or first natural frequency; (b) any of these under different constrains, such as: maximum deflection, maximum stress, target weight (volume), target stiffness (displacement) and durability. The choice of design variables ranges from geometrical parameters, control points of spline functions [16], position of nodes [17], shell thickness [18], and beam cross-section [19], to angle of fibers from compound materials, etc. As design variable restrictions a designer can have: upper and lower limit of the design variables (fixations, limitations), discrete and continuous. Also symmetrical conditions and constraints for manufacturing conditions (drilling, casting or forging) are also possible. Particularly, two kinds of structural optimization are often used: topology optimization and shape optimization [20].

Topology optimization is used to find the optimal topology of a part by describing an available space for the part using a FE model, while the optimization algorithm helps find an optimal material distribution. Element properties such as density and Young’s modulus from the FE model are changed during the optimization process until an optimal shape is achieved. After defining a topology for a shape optimization problem, a common practice is to use a fixed set of shape variables to describe the design boundaries [21]. The values of the shape variables are then optimized to provide the lightest possible structural member. Topology optimization is used to reach the highest ratio between volume and some geometrical parameters. It determines the optimal material distribution within a given design space. For example, it takes out the elements under low stress in geometry by modifying the apparent material density, considered as a design variable in a FEM model. A basic FE model is created and analyzed in a design area with given boundary conditions. Commonly, the aims are to maximize stiffness or maximize the natural frequency of a product. The constraints of the design are the following: the fixations, material volume and largest displacement allowed. The design variables are the material density of the elements, which are counted commonly in hundreds of thousands, which means a huge number of design variables. The goal is, given a predefined design domain in the 2D/3D space with structural boundary conditions and load definitions, is to distribute a given mass, which is a given percentage of the initial mass in the domain determined, in such a way that a global measure takes a minimum (maximum) value. This type of topology variation is being analyzed only as a reference and basis for the kind of optimization that is going to be derived in this dissertation.

Shape Optimization consists of changing the external borders of a mechanical part [22]. The geometry of the product is defined in terms of surfaces and curve parameters that define the outer boundary of the product, and allows more freedom for manipulation. Here, the topology remains unchanged. The shape of the structure is modified by the node locations of a product modeled with the finite element method (FEM). The aims are to decrease the stress or the volume or maximize the natural frequency. Constrains to the design include fixations and restrictions for displacement of part borders. The design variables of the product for geometric models are length, angle and radii measurements; and for the FE model, node coordinates. After defining a topology for a shape optimization problem, a common practice is to use a fixed set of shape variables to describe the design boundaries [23]. The values of the shape variables are then optimized to provide the lightest possible structural member. Specification of the initial set of shape variables is done while maintaining accurate structural analysis predictions by automating the variable selection process.

Another kind of optimization is an advanced form of shape optimization is known as topography, in which, e.g., a distribution of ribs and pattern reinforcements is generated in a specific design region. The approach in topography optimization is similar to the approach used in topology optimization, but shape variables (node coordinates of a FEM model) are used instead of density variables [24].

Other approaches for structural topology optimization present a method based on level set models for optimizing linearly elastic structures which satisfy a design objective and certain constraints. In this method, the structure under optimization is implicitly represented by a moving boundary embedded in a scalar function (the level set function) of a higher dimensionality [25]. Other cases of topology generation—that is, establishing a feasible configuration to meet given functional requirements and size and shape optimization—to meet the prescribed quantitative performance can be found in micromechanics as well [26]

Each optimization method uses a strategy to get the optimum of the objective function. The choice of the optimization method and the strategy depends mainly on the properties and number of the variables, the objective functions and constrains, and how these are used in the optimization. Specific criteria for optimization problems include: the number of variables (often many of them); characteristics of the objective function (continuous, discontinuous, linear/

quadratic/arbitrary, etc.); restrictions characteristics (none, several, etc). Moreover, the external conditions for choosing an optimization method rely on the required accuracy (improvement or exact optimum); efficiency of the algorithm; computing time and memory space; user friendliness and complexity of the problem formulation.

Current CAD/CAE packages are conceived to help parametric variations of the parts. In recent years, shape and topological optimization have been introduced into CAD/CAE environments [27]. However, the shapes obtained this way are not structured CAD models and need manual post processing or even redesign to convert them from mesh or neutral geometric models to full CAD models.

One of the aims of this research is to further develop the optimization systems mentioned above by adding new concepts to the previous paradigms. In doing so, a different kind of optimization is inferred by taking the characteristics of the last three FEM based types just mentioned. To get a similar behavior (shape, topology and topography) within a CAD model, the geometry of the product is described in terms of splines. The "splining" approach extends these features, allowing the introduction of innovative concepts [28]. This approach is explained further in the next sections. Evolutionary design takes both CAD and CAE, and extends the analysis and detailed design capabilities for which they have been used until now. Moreover, the processing capabilities of computers continue to grow, which makes the resolution of more complex FEM models possible.

The possibility of achieving changes of geometric shapes of whole parts and features directly in the context of representation with commercial CAD packages needs new concepts. Tools for facilitating automatic shape variations in parametric 3D CAD environments with the purpose of using genetic and evolutionary algorithms for the shape and topological optimization are presented in this work. The central idea here is that generating creative designs would only be possible by going beyond the bounds of a representation, and by finding novel solutions, which simply could not have been defined by that representation. The concept of Generative CAI Process is derived.

## 1.3 Shape Optimization Based on Genetic Algorithms

The shape optimization of engineering objects is an important part of the design process [29]. Many different approaches have been adopted in this area, from the homogenization method [30] to the topological derivative method [31]. Shape optimization based on genetic algorithms (GA), or based on evolutionary algorithms (EA) in general, is a relatively new area of research. The foundations of GAs can be found in a few articles published before 1990 [32]. After 1995 many articles on investigation and applications have been published, including a great number of GA-based geometrical boundary shape optimization cases. The interest in research in evolutionary shape optimization techniques has only recently started to grow, including one of the most promising areas for EA-based shape optimization applications: mechanical engineering. Genetic algorithms are a computational tool for the search for optimal design solutions based on the selection of the "fittest", even when the internal mechanism of the system under optimization is not well known. Genetic algorithms have been used in different applications; one is the optimization of geometrical shapes, a relatively new area with high potential for research.

There are applications for shape determination during the design of machine components and for optimization of functional performance of these components, e.g. antennas [33] [34], turbine blades [35], aircraft geometries [36], etc. Difficult shape optimization problems are common in many other areas as well. In the future, one of the areas with the most potential for EA-based shape optimization applications is mechanical engineering, as designing machine components typically includes shape determination and optimization for functional surfaces of the components. In the field of mechanical engineering, methods for structural and topological optimization based on evolutionary algorithms are used to obtain optimal geometric solutions that were commonly approached only by costly, time consuming iterative processes. For example, a paper published in 2004, reported that an efficient optimum design process was developed and applied to systematically design a lock-up clutch system for a torque converter used in an automatic transmission [37]. A simulated annealing algorithm was applied to find the parameters of the compressive helical damper springs in the clutch. Next, FE-based shape optimization was coded to find the shape of the clutch disk that would satisfy the strength, noise and vibration requirements. Using the optimization algorithm, parametric studies were performed to see how spring diameters and frequencies of clutch systems changed as the damper spring traveling angles and the torques were varied. A genetic algorithm for spring optimization and an FEM-based algorithm for optimizing disk shape were utilized in the project reported in that paper.

The shape optimization approach has been also employed to optimize forging preforms using the reverse simulation method. Bramley developed a reverse method in which a preform shape was obtained by reversing the direction of the velocity field to give a minimum value for the overall rate of energy dissipation and which was optimized by the current boundary contact conditions [38]. The approach employs a criterion based on the concept of the material distribution to determine the boundary condition and to evaluate the complexity of the deformed shapes resulting from reverse simulations [39].

Genetic algorithms to optimize existing designs by parameterization was the first type of evolutionary design to be widely used. GAs integrated into CAD and CAE systems are used in parametric and structural optimization to find optimal topologies and shapes of given parts under certain conditions. Advanced CAD and CAE software have their own optimization capabilities, but are often limited to some local search algorithms, so for this dissertation the decision was made to use genetic algorithms. To make such integration, it is necessary to develop an interface to link the GAs to the CAD models and to the CAE analysis. This dissertation presents an approach to this task and some approaches that can be used to construct a strategy on crankshaft design and development. The approach of evolutionary design with splines [20] has the potential to be generative.

## 1.4 The Splining Approach

In the mechanical field, geometric parameterization is used to define the kind of changes described by design variables. The geometry of a product keeps its topology but changes in terms of its dimension values or some properties of the structural elements, such as thickness, width and height. Parameterization of the dimensions that describe the product allows optimization based on measures, and is performed through CAD geometric variations. Shape optimization determines the optimal boundaries of a structure for the given fixed topology. Design variables are typically spline control points defining the shape of a structure in 2D or 3D [4].

An efficient way of creating and controlling the form parameters of a given geometry is to convert them into spline models. The spline curve entity is a non-uniform cubic spline, defined by a series of three-dimensional points, tangent vectors at each point, and an array of un-normalized spline parameters at each point. A spline curve is a sequence of curve segments interconnected to form a single continuous curve. Curvature continuity is an important requirement for many applications, and most shapes are simply too complicated to define using a single curve. A B-spline (the most popular spline) is a series of Bezier curves. A Bezier curve always passes through the first and last control points and lies within the convex hull of the control points. A B-spline of degree $n$ can be defined with a knot vector. A knot vector is a list of parameter values, or knots, that specify the parameter intervals for the individual Bezier curves that make up a B-spline. A B-spline curve with an evenly spaced knot vector is known as a uniform B-spline. If the knot vector is not evenly spaced, the curve is called a non-uniform B-spline. B-splines are said to have the property of local control, since any given control point can exert influence at most $n$ curve segments [40].

A spline is commonly used to define complex geometries that cannot be modeled, or are difficult to model using lines, curves, circles, etc. Shape parameterization is based on different kinds of curves. B-splines [41] are used to define curved profiles. The advantage of these spline geometries is that they have a shape that is easily controllable through control points. Control points are points in space through which a curve must be adjusted. Another characteristic of these curves is the possibility of making special adjustments (like sharp angles), depending on the relative position of the control points. The splines can be applied to define geometry of complex shapes, such as fan wings [42] and also can be used in mesh representation for FEM models [43]. In other words, the parameters of a spline control its shape, so any kind of transformation can be executed.

The design of parts in 3D parametric CAD packages is based on sketches of the features that determine the topology of the model´s parts, which are related to each other through different Boolean operations. In the parametric sketches, common basic geometric elements such as lines, polylines, arcs and circles are used. As splines are geometric entities that are very well suited for shape variations, for more complex geometries, splines will be used. The main idea of this approach consists of converting all the basic geometric elements of the sketches that constitute a part, into spline features, whose shape is controllable by manipulating the coordinates of its control points. Splines are used to define curved bodies and profiles. A circle, for instance, could be easily defined by a spline by placing all of its control points on the circle perimeter. Arcs and straight lines are also easily replaceable for splines. To obtain higher levels of freedom in shape variations, basic geometric features that only allow parametric variations are converted into geometrically equivalent spline models. This operation has been named “splining” [28].

The increased flexibility derived from the splining approach [44] could suggest a very high number of variation possibilities for a given part, including different shapes for features; even though genetic algorithms are being studied for their ability to decide automatically the best-suited modification for a given purpose, there are many initial possibilities.

The “splining” approach for optimizing designs using GAs is relatively new. It has been used, for example, for data fitting using genetic algorithms [45]. Practitioners of evolutionary optimization using splines commonly start the process with an existing design, substitute the current construction with splines, as the number of degrees of freedom can be increased at will by adding control points, and then parameterize the control points of the splines that embody those parts of the design they feel need improvement. The control points are encoded as genes, and the alleles (values) from which the parameters are described are evolved by an evolutionary search algorithm, e.g. Genetic Algorithms. In the field of mechanical engineering, this approach has been applied to the computer design and optimization of the profile shape of a cam for a diesel engine [16].

The goal of the cam optimization was to optimize the movement of the system. That article describes a genetic algorithm with a computer-aided approach for preliminary design and shape optimization of cam splined profiles for cam-operated mechanisms. The primary aim of the project was to create a complete systematic approach for preliminary cam shape design including design automation and true cam shape optimization with respect to the simulated computer models of cam mechanisms. Shape optimization of a cam cross-section is a multi-objective optimization problem of a two-dimensional geometric splined shape in a constrained environment. The conclusions of the article are that with genetic algorithms it is not only possible to design and optimize geometric shapes, but also to develop procedures and tools to automate the design process and get the best form simulation software better and more effectively than what an experienced person could do by trial and error.

Splining may constitute a basis for allowing crossbreeding of different geometries, as the points of the splines may be used as “genes” from the parents that transmit their properties to their offspring. Geometric crossbreeding should be done at the sketch level of the construction features of the parts’ geometry. The possibility of crossbreeding two or more parts or assemblies perhaps would need to first, recognize when two different parts are able to generate an offspring, or when two different assemblies generate a new assembly. At feature level, the crossbreeding is possible between two different geometries of sketched features, see Figure 1-1. However, at the part level some similitude inside the tree of the parts’ features should be present to allow two different parts to crossbreed. The same may be true for crossbreeding two different assemblies; i.e., some similitude of the assembled structure could be required for achieving a “successful” crossbreed of two assemblies or subassemblies.

Sketches are controlled through its parametric structure, which changes during the splining approach. A solution is found for substituting the initial parametric structure with a new controlling structure that allows parametric similar control of the size of parts and features being analyzed. This solution is accomplished through the control points of the spline.

Even though the focus on the splining technique is about shape variations that allow the persistence of CAD models, there are other interesting ways of obtaining these same advantages by developing special shape variations at mesh level and then translating them into direct change for a previously “splinized” CAD feature. For this purpose mesh morphing in existing commercial finite element meshing software exists, which allow shape variations to be made in a mesh model without remeshing it.

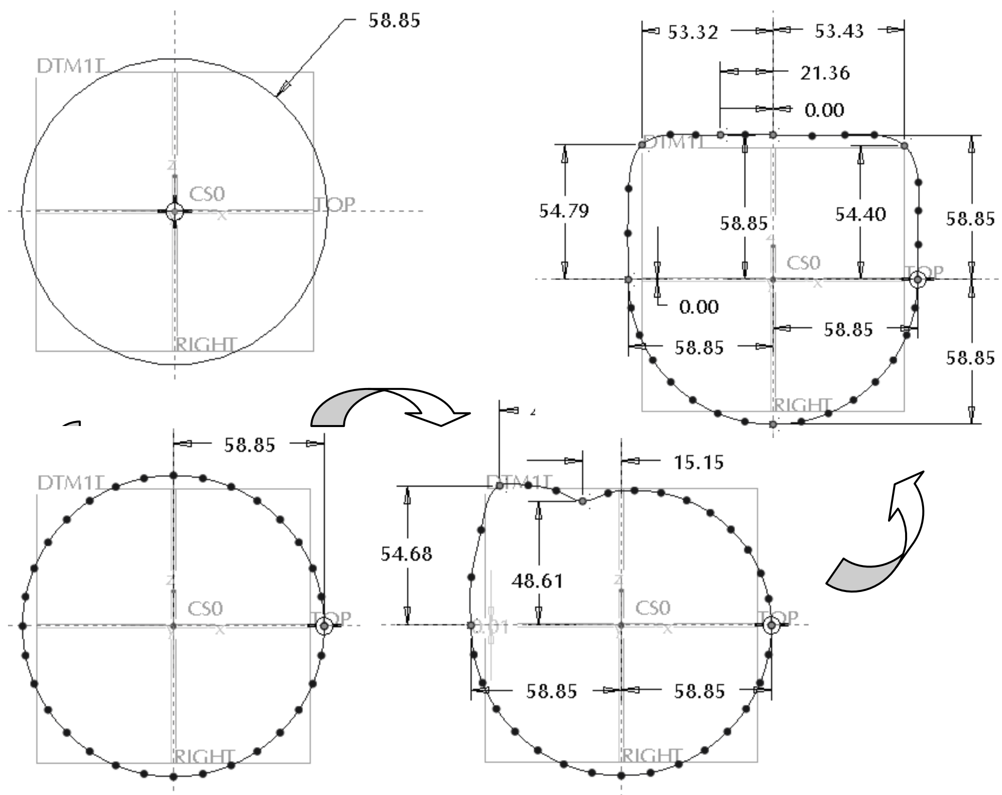



Figure 1-1 Splining of a circle for better geometry manipulation.

The designs are often judged by making an interface of the system to simulation software or analysis software, which is used to obtain a fitness measure for each design. These tools result in an extended optimization method, achieved by representing shapes with splined curves [11]. This eliminates non-parametric dependency on FEM and allows a concept variation of forms. Automatic variations in shapes are produced by the computer and concept variations are evaluated by analysis software.

## 1.5 Justification, Hypothesis and Objectives

Structural optimization has achieved a level of development that makes it possible to go beyond the current limits of optimizing shapes of parts. This kind of optimization also allows designers to improve the traditional design process, helping them to make decisions regarding new innovative shapes and solutions that arise during the shape variation and simulation process. Designers can now take advantage of not only using parametric CAD models to analyze the influence of geometric parameters variations, but also the effect of shape variations on the required performance of their designs. This thesis presents a strategy that allows the designer to perform an automatic search process for crankshaft design objectives based on genetic algorithms, integrated with the simulation of crankshaft behavior with respect to its balancing and torsion vibrations. At the same time, it looks for an optimal forging process, one that allows improve the crankshaft design process, compared to the design approaches used until now. An important characteristic of the strategy presented in this thesis is the automation of the search process, as genetic algorithms through crossbreeding and mutations of the counterweight profile control the variation of shapes in search for the target balance. At the same time, seek to achieve a target position of the crankshaft's center of gravity and achieving an optimally forgeable shape.

### 1.5.1 Justification of the research

The motivation for the development of this effort is to contribute to the forging industry by the use of the breakthrough optimization techniques, which improves the design of new products. The intention is also to bring to the world of technology research on the use of evolutionary techniques in a mechanical design problem as the forging design of a crankshaft is. An opportunity has been detected to carry out the research using an industry-academia approach. Eight years of experience in Forja de Monterrey have made the author acknowledge needs that can only be addressed by this kind of project to achieve improvement benefitting the country, the industry and the academy research. The next following two quotes express the perceived trend in the industry motivating this research dissertation:

"In the year 2020, forging will be the cost-effective, preferred process by which metal components of superior quality, integrity, and performance are produced for critical and demanding applications." (Technology Initiative, Forging Industry Association, 2002. http://www.forging.org/ techno/Vision.htm#execsum)

"To maintain the main development thrusts of the forging user industries, forging companies need to enhance their take-up of new technology and in particular the use of modeling techniques for part and process design". (Modeling tools for the forging industry - FORGE-NET. The European Commission. http://europa.eu.int/comm/research/brite-eu/thematic/html/1-1-07.html)

Optimization is a design activity where the mean is shifted to target and reduce the variability for all the responses [46]. Optimization is used to take a particular design and improve it to a level of superior performance, but once this level has been achieved, further improvement is impossible through the use of the same optimization tools because the best combination of parameters in the design has been found. In some cases, the way designers formulate their problems can lead to excessive assumptions, engineering approximations, and restrictions, with which an optimum design is created, but in a sense that it is the only feasible design, not the best solution. To overcome this apparent contradiction, this research proposes a process that integrates shape variations and supports the use of interesting optimization approaches, e.g. genetic algorithms. All of this takes place in an automated environment with the ultimate goal of using computers to help designers to overcome design fixation when developing a product. And then, due to this combination of factors, inventive solutions may arise by the emergence of ideas from the stimulated mind of the designer, which can be then regarded as generative computer aided invention. From the last, the research is intended to prove the next:

### 1.5.2 Hypothesis

*The integration of 3D-CAD parametric models, adaptive splinization, CAE analysis and genetic algorithms, by means of the deployment of tasks that automate the design process, improves the performance and manufacturability of engine crankshafts. The achievement of novel responses that emerge during the genetic evolution process can lead to new generative computer aided inventive solutions that help overcome design fixation.*

*The optimization capabilities of computers can be extended, when introducing elements of inventive methodologies, in a way that makes them an innovation tool for designers. This will allow them to get the best from simulation software, and reach novel solutions during product development of crankshafts more efficiently than by the trial and error method.*

For the demonstration of the hypothesis, the research was conducted within the following:

### 1.5.3 Objectives

a) The development of a strategy that combines the activities of concept generation and evaluation; and merges them in a loop inside the product development process, in order to improve the design of a crankshaft. The strategy aims to integrate the loop into a computer aided invention framework, and by doing so, extend the optimization capabilities of computers to provide inventive results.

b) The generation of novel designs of the case study of an engine crankshaft, in order to obtain an inventive solution that represents an improvement of the crankshaft, in terms of imbalance, dynamic response and manufacturability. By using the genetic algorithms for spline manipulation, it is possible to monitor the progresses of the shape construction as real-time identification of design trends and make changes to the definition of the shapes.

Under those objectives, the research is aimed to improve the traditional design process to become a computer aided invention process. Designers can take advantage of the methods and make use of parametric CAD models available during the product development process to analyze the influence of geometrical variations. An important characteristic of the tasks presented in this thesis is the automation of the optimization/innovation process. From the geometric model modifications to the evaluation of the analyses, modern CAD software and optimization tools meet all the requirements for such evolution.

## 1.6 Conducted methodology for the completion of the research

The methodology conducted in order to reach the objectives and for instance, demonstrate the hypothesis, consisted of the following steps.

A) Problem statement. It is analyzed the functioning of crankshafts and its geometrical construction. The shapes that are going to be converted to splines are selected and it is decided over the set of parameters to be controlled.
B) Literature review. It is conducted a literature review of the methods and tools that have been used in previous approaches. The elements of those methods that have the necessary conditions for integration are selected to be used in the research.
C) Development of the strategy. The computer aided invention strategy is development with the elements selected but integrated by a set of tasks. Those tasks are defined by the kind of optimization that is pursued.
D) Deployment of the implementation. The strategy and particularly the set of tasks are implemented into the case study of the crankshafts. The results of the improvement of the crankshaft in terms of performance are obtained.
E) Construction of the theoretical foundations. The basis of a computer aided invention strategy are presented. The fundamentals of a framework to extend the product development cycle into an innovation approach are explaided.

The previous steps of the methodology for the completion of the dissertation are presented and explained in the following order of chapters.

Chapter 2 states and explains the problem of optimizing an engine crankshaft. The case study selected and how the CAD modeling of the crankshaft is made using splines are presented. A brief introduction to the dynamics of a crankshaft, and the imbalance and its importance during the manufacturing process are included. The chapter describes the equations of imbalance and the initial measurement to be used as reference for the improvement during the development of solutions.

Chapter 3 offers a review of the literature, focusing on general optimization methods and, particularly, on genetic algorithms. It is explained why a stochastic technique as the geometric algorithms are necessary in the research. The multi-objective optimization strategies and why the case study was considered suitable for multi-objective analysis are also discussed. The concept of Pareto frontier is explained and how the Pareto graph can be used to reveal conflicts in performance parameters of a system. A description is given of the optimization software DAKOTA and its code structure.

Chapter 4 present the overall strategy developed for extending the optimization paradigm into an innovation framework, and a description of the different tasks that are integrate. The first task consists of the optimization of the imbalance of the crankshaft, taking into consideration only geometric modifications of the CAD model. The second task extends the use of CAE tools to analyze the dynamic behavior of the crankshaft by measuring, together with the imbalance, the eigen-frequencies of the crankshaft.

Chapter 5 describes the deployment of the tasks, with the case study of the crankshaft, introduced in chapter 4 and presents the results obtained from each. It is made a detailed description of the geometry substitution by spline curves of the counterweights. It is explained the construction of the CAD model and the introduction of the imbalance formulas inside the CAD model thanks to the software functionality. It is described how the optimization loops are built to have the different software integrated via programming in Java. The results of the tasks are presented in the form of modified crankshaft geometries.

Chapter 6 gives a general description of the optimization/innovation approach. A proof of concept of the results from chapter 5 is performed using simulation of the forging process. It is presented a reflection of how the strategies are more than an optimization approach and extend the optimization capabilities of computers. It is explained how the combination of tools allow to conduct an innovative solution by overcoming conflicts in a multiobjective problem. It is also presented the ideas relating the TRIZ integration with Genetic Algorithms.

Chapter 7 contains the conclusions and future work that can be derived from the research that is developed in this dissertation.

# Chapter 2 Problem Statement

The case study was selected in view of the importance of motor crankshafts in the automotive industry [47] and the increased performance requirements for engines, which have raised the production of forged steel crankshafts worldwide. The auto industry will continue to move toward forged crankshafts, as engines become more fuel-efficient, producing more horsepower per liter. During the late 1970s, designers recognized that the need for greater fuel economy would be driving up the engine horsepower per liter through greater compression ratios and much higher rpms. Since roughly the mid-1980s it has been recognized that to achieve those goals, designs would have to be changed to achieve stronger crankshafts. Particularly, the development of engine crankshafts is subject to a continuous evolution due to market pressures. Fast market developments push the increase of power, fuel economy, durability and reliability of combustion engines, and call for reduction of size, weight, vibration and noise, cost, etc. Optimized engine components are, therefore, required if competitive designs are to be attained. Given these conditions, crankshafts, which are one of the most analyzed engine components, must be improved [47]. For crankshaft material, the choices were largely between austempered ductile iron (ADI) and forged steel. Since the low-cost nodular iron would now have to change to a heat-treated product cast from a special grade, the economics of castings were not as positive in comparison with forgings. Furthermore, the advent of micro alloy steels allowed forge shops to compete effectively with ADI because micro alloys can be used to eliminate the heat treating cycles for forgings [48].

An important direction in improvement trends is the optimization of its geometrical characteristics. In particular for this paper, imbalance, first Eigen-frequency and the forge-ability are the optimization objectives. Analytical tools can greatly enhance an understanding of the physical phenomena associated with the above mentioned characteristics and can be automated to do programmed tasks that an engineer requires for optimizing a design [2]. Imbalance, one of the quality parameters of crankshafts, has a significant impact on the life of the entire system. In the forging process, given the variations of material composition, temperature, etc., the imbalance dispersion increases compared to the casting of crankshafts. It is of particular importance to reduce this variation right from the conceptual design of the crankshaft [49] [50]. As the crankshaft is an element with different functionalities (transform linear movement into rotation, manufacturing) and physical behaviors (natural frequencies, imbalance) it is especially suitable to be case study for an optimization problem. Crankshafts have performance parameters and restrictions that are commonly analyzed and studied, with software tools, during the design and development process of an engine. These performance parameters are to be optimized (minimized or maximized) by making variations in the geometry.

The selected goal of the crankshaft design study for this research is to reach the imbalance target and reduce its weight and/or increase its first Eigen-frequency. The design of the crankshaft is inherently a multi-objective optimization (MO) problem. The imbalance is measured in both sides of the crankshaft so the problem is to optimize the components of a vector-valued objective function consisting of both imbalances [51]. Unlike the single-objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto-optimal set. Each point in this set is optimal in the sense that no improvement can be achieved in one objective component that does not lead to degradation in any of the remaining components [36].

The objective functions of imbalance are also non-linear. Auxiliary information, like the derivatives of the objective function, is not available. The fitness-function is available only in the form of a computer model of the crankshaft, not in analytical form. Since the approach required taking the objective function as a black box, and only the availability of the objective function value could be guaranteed, no further assumptions were considered. Thus, the only realistic alternative was applying a stochastic global optimization approach. Since GAs have already widely demonstrated capabilities for effective, efficient and robust global optimization in cases for many black-box type computer models, including many shape optimization models, GAs were considered the most attractive alternative for the research purposes. The Pareto-based optimization method, known as the Multiple Objective Genetic Algorithm (MOGA) [52], is used in the present MO problem, to find the Pareto front between these two fitness functions.

## 2.1 State-of-the-Art in Crankshafts

A crankshaft used in a V-type six-cylinder engine is equipped with six crank pins (numbered P1 to P6, see Figure 2-1), four crank journals (numbered J1 to J4), and nine arms (numbered A1 to A9) for connecting the crank pins and the crank journals to each other. Some of these arms are each equipped with a counterweight (numbered W1 to W9) to reduce the rotational force of inertia generated in each arm and the unbalance couple generated in the crankshaft.

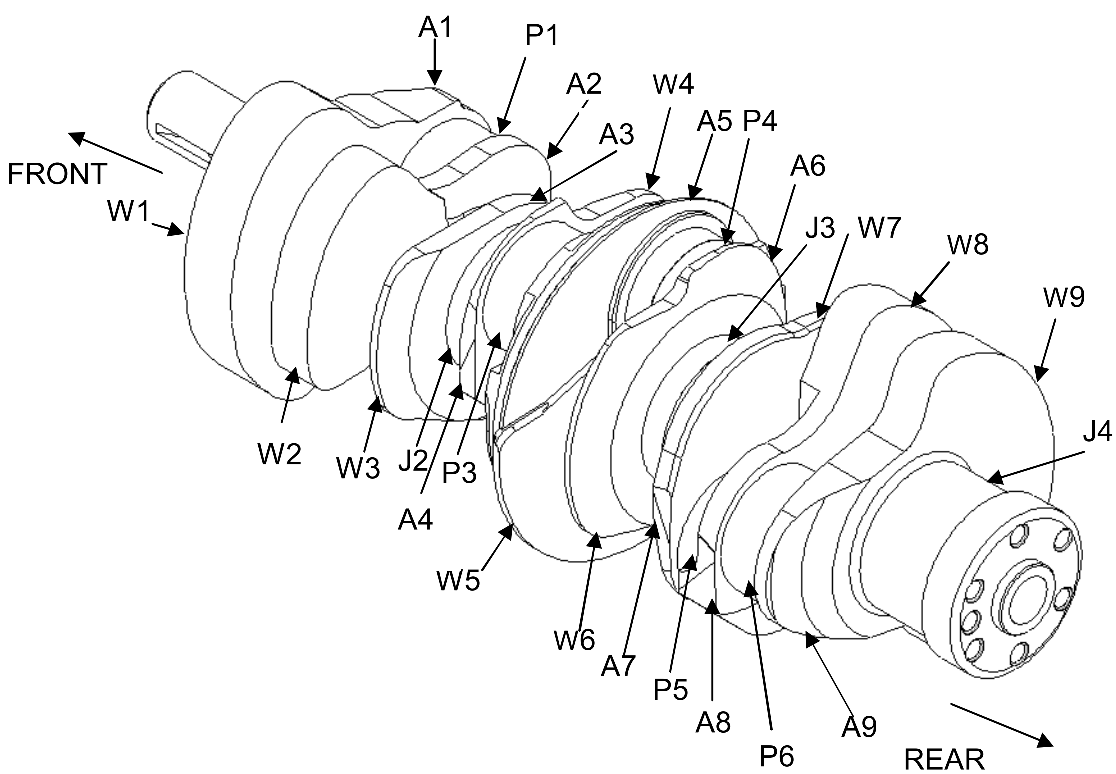


Figure 2-1 Perspective view of an engine crankshaft

In a conventional crankshaft, the counterweights are situated in a direction completely opposite the crank pins connected to the arms with respect to the central axis of the crankshaft. That is, in the case of an arm to which only one crank pin is connected, the counterweight is situated in a direction completely opposite that crank pin (see Figure 2-2). In the case of an arm to which two crank pins are connected, the counterweight is situated in a direction completely opposite the midpoint of the two crank pins.

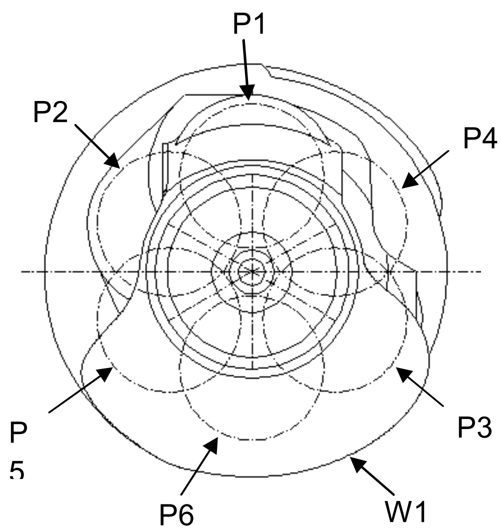



Figure 2-2 Front view of an engine crankshaft

In a review of the history of the development of V6 type crankshafts, an exhaustive patent search was conducted. The original V-6 Engine can be found in a patent [53] from the former General Motors Corporation. The text of the patent is very descriptive with respect to the intention and goals of the research:

“The present invention relates to internal combustion engines, and more particularly relates to the crankpin arrangement and balancing means for a 60° V-6 engine.

“Modern automotive internal combustion engines are required to operate at high speed with a minimum of vibration. This requirement necessitates accurate and complete balancing of the inertia forces and couples set up by rotating and reciprocating masses within the engine.

“V-6 engines are rigid and compact and are therefore desirable as automotive power plants. The lack of a suitable crankshaft design and balancing means for these engines has delayed their adoption by the automotive industry. It is well-known that if the forces and couples of an engine are not balanced, excessive vibration will result. This vibration greatly increases the wear on engine parts, and passenger discomfort and reduces the life of the engine. For these reasons, it is necessary to reduce engine unbalance to a minimum.

“Engines may be out of balance, with resulting vibration and excessive wear, due to either unbalanced forces or unbalanced couples. The unbalance forces may be "primary" or those operating at engine speed or may be higher order forces which are classified as a "secondary," "tertiary," etc. corresponding to the harmonic of engine speed at which they are effective. The term "unbalance couple" as applied here, is the product of the force times the distance the force acts from a preselected datum line. These couples may be either so-called "shaking" couples which are manifest by forces acting in a single plane lying in the longitudinal axis of the engine or they may be rotating couples which change their direction of application with respect to the line of center of the engine either at engine speed or at some harmonic thereof. The design of the present engine eliminates all shaking couples. The unbalance, due to rotating forces resulting in rotating couples of higher order than secondary, are negligible in the present engine. The primary unbalance forces and rotating couples are counterbalanced by the use of weights located on the engine crankshaft and fly-wheel.” See Figure 2-3.

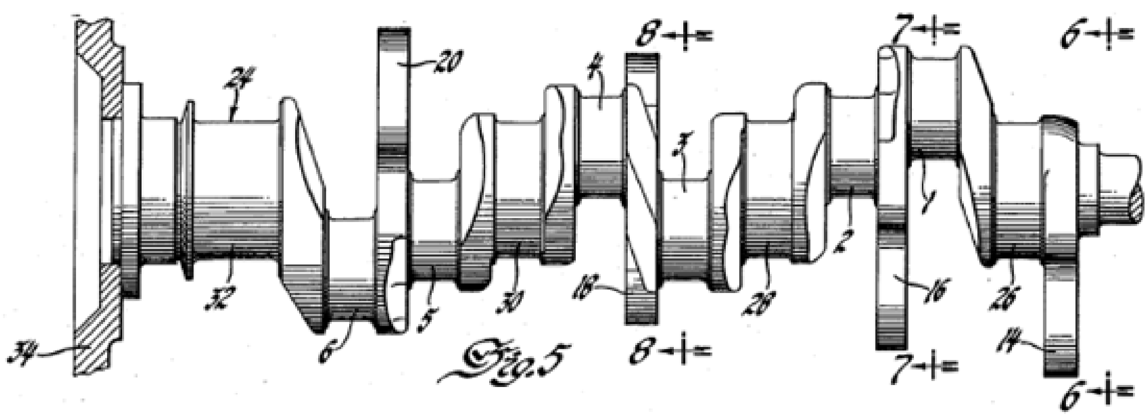


Figure 2-3 Crankshaft from Original V-Six Engine General Motors Corporation.

In the patent databases, the older the patent, the simpler the counterweight’s profiles; crankshafts are modeled with the use of arcs and particularly the use of symmetry. It means that newer crankshafts are less symmetric, in order to adapt to the balance, but still use simple arcs.

A patent from 1985 [54] provides the crankshaft with two more counterweighs than the original crankshaft, which consisted of counterweighs in the extreme sides. The crankshafts then have counterweights in the first, second, eighth and ninth arms. A patent from 1988 [55] claims to provide an excellent engine crankshaft structure which is useful in the drive system field in the automotive industry. It uses the conventional crankshaft structure that it can well afford to bear large surface pressure and to resist a large bending and shearing stress and by minimizing the number of counterweights. Patent from 1996 [56] presents a crankshaft of a V-type 6-cylinder internal combustion engine which claims to ensure a decrease of the engine vibration and surface pressure affecting each crank journal by achieving effectively dynamic balance of a whole crankshaft and dynamic balance couple between inner main bearings with an optimum size of balance weight. A patent from 1998 [57] involves a crankshaft in which counterweights are provided respectively on only the first, second, fifth, eighth and ninth arms, with the weight and thickness of the first to fifth counterweights being set to predetermined values respectively.

The next two patents are the newer regarding V-type 6 cylinder engines. A 2007 patent [58] is an invention to provide a crankshaft for a V-type internal combustion engine of the type with a pair of counterweights arranged outwardly of the axially outermost journals. The crankshaft includes a first to a sixth crank pin, a first to a fourth main journal and a first to a ninth crank arm. At least the first, the second, the eighth and the ninth crank arms have respective integral counterweights for balancing the crankshaft. To reduce bending stress on the areas where the second and the eighth crank arms are connected to the respective pairs of adjacent crank pins (bending problems particularly present in the forging dies during the manufacturing process), the axial dimensions of the first and the ninth crank arms are reduced. Being substantially the same as those of the adjacent crank arms (second and eight); in order that the axial length of the two end spans of the crankshaft is the same as that of the intermediate span. In order to set the balance of the crankshaft back to the original value, two additional counterweights are arranged outwardly of the end journals and their axial dimensions are comparable to those of the end crank arms, see Figure 2-4.

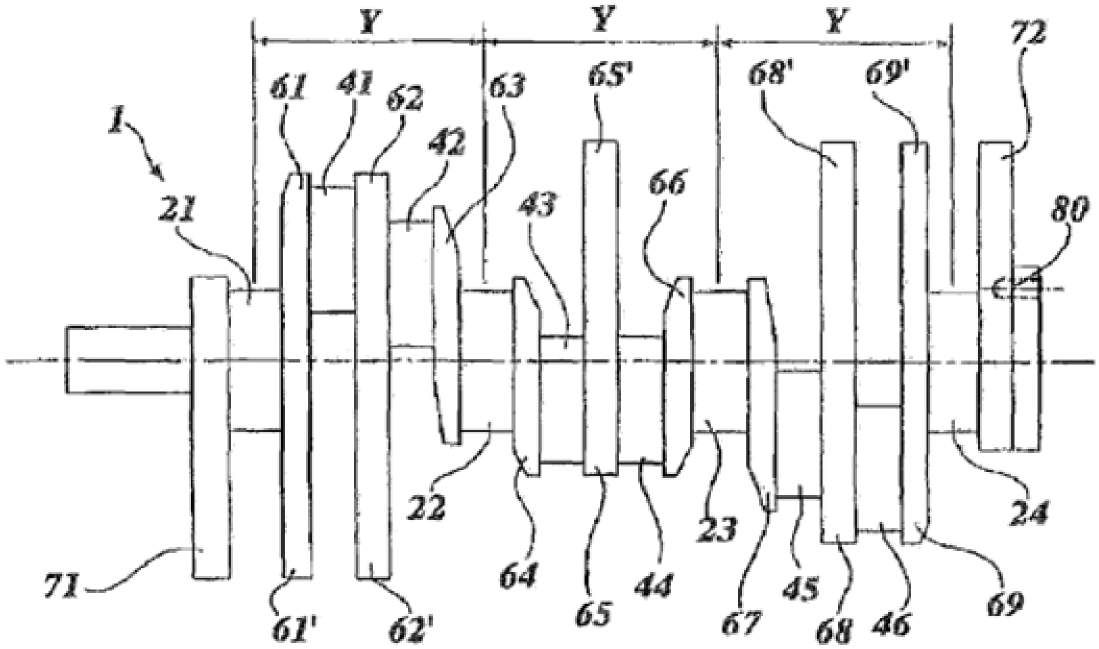



**Figure 2-4 Patent US 7,210,373 B2, May 1, 2007**

Another patent granted in 2007 [59] is for an invention to provide a crankshaft for a V-type six-cylinder engine capable of reducing the generated unbalance couple to a sufficient degree. Thus, in the crankshaft the arm A5 has no counterweight, which makes it possible to achieve a reduction in weight. Without counterweight 5 the effect of reducing the bearing load generated in the crank journals J2 and J3 through the rotational force of inertia of the crank pins P3 and P4 is lower than that in a case of a crankshaft in which the arm A5 has a counterweight. Accordingly, in the crankshaft 1, which is for use in a diesel engine, there is practically no disadvantage in increasing the bearing load due to the fact that the arm A5 has no counterweight, making it possible to attain, substantially and exclusively, compatibility between a reduction in weight and a reduction in unbalance couple. Furthermore, arms A1 through A3 and the arms A7 through A9 respectively with counterweights, are concentrated on both ends of the crankshaft. So that if the total weight is the same as that of a construction in which counterweights are provided on arms near the center (e.g., a construction in which each of the arms A1, A2, A4, A6, A8, and A9 has a counterweight), a moment generated is larger. Whereby it is possible to more efficiently achieve a reduction in unbalance couple. See Figure 2-5.

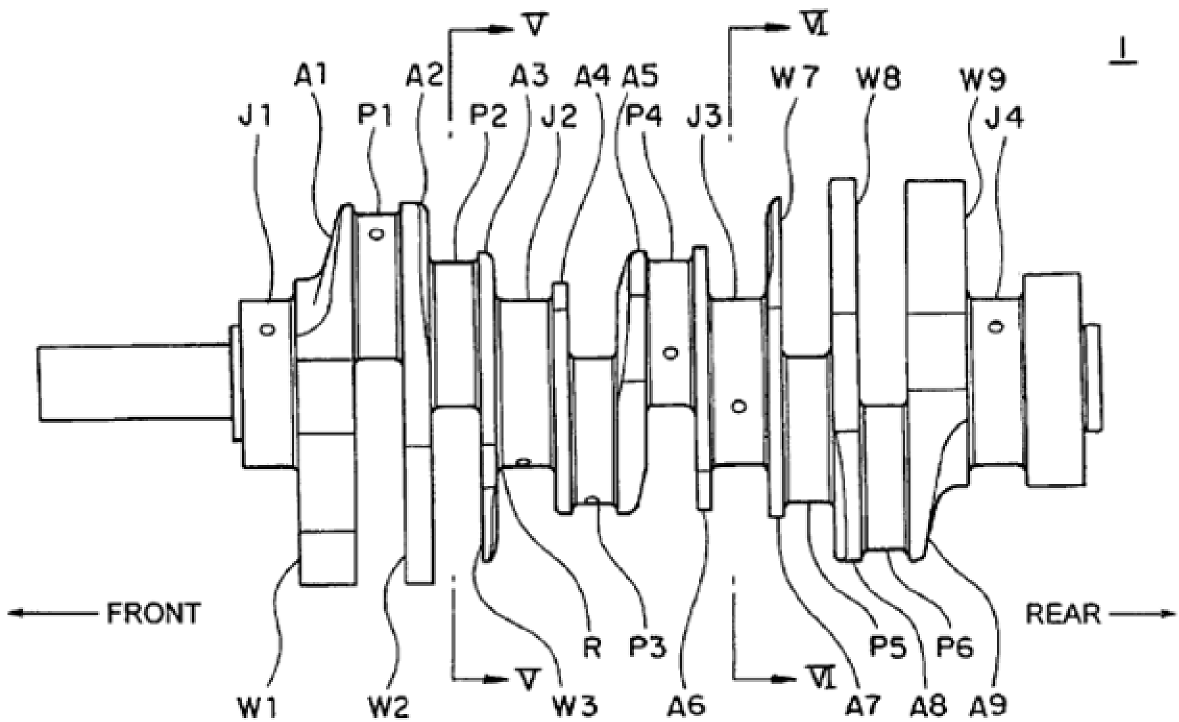



Figure 2-5 Patent US 7,234,432 B2 June 26, 2007

A patent from 2001 [60] describes a crankshaft not for a V6 engine, but rather for a forged crankshaft for an internal combustion engine, and more particularly to a crankshaft with larger counter weights, but which does not cause interference between the counter weights and neighboring parts of the engine. According to one aspect of the invention, an engine crankshaft with at least one counter weight made by forging is provided. As the forging being performed by pressing two dies to each other, it is characterized by the fact that the counterweight has a plurality of inclinations to facilitate removal of the dies from the crankshaft after the forging process is completed. The degree of inclination of the plurality of inclinations is reduced since the plurality of inclinations is away from a die-mating surface. However, the plurality of inclinations refers to the width of the counterweights, not to the profile, which is not claimed.

From the point of view of the particular crankshaft part under this study (V-type 6 cylinder), the geometry has commonly consisted on arc-shaped counterweight profiles. The forging process demands a better flow of the metal into die cavity and robustness in the balance and dynamic behavior of the crankshaft. To solve this task, therefore, not only expertise is needed, but also new strategies. In this research, the aim is to develop a method that brings the designers new alternatives. In this thesis, the shape optimization of a crankshaft is discussed, with focus on the geometrical development of the counterweights.

## 2.2 Crankshaft Dynamics

The case study presented here involves the development of an engine crankshaft, with emphasis on its imbalance and dynamic behaviour (i.e. Eigen values). The goal of a balance analysis of engine crankshaft is to develop the geometry so that a specified previous imbalance is obtained on two correction planes [61]. These two planes lie where the outermost counterweights are located, and so the imbalance could be corrected at the manufacturing shop by performing specific drilling on those counterweights.

### 2.2.1 Balance basic concepts

The crankshafts are special cases of the asymmetrical rotor, since more than two counterbalances are needed to correct balancing [51]. The configuration of a crankshaft with multi-counterbalance can be used for the purpose, and each counterbalance has a limit where material cannot be removed. However, any two counterbalances can be taken in combination as a set of balancing planes. Thus, a complete correction in balancing crankshafts is usually not finalized by using only two counterbalances; and all possible combinations of counterbalances may be needed.

In other words, such previous imbalance represents a value of mass by radial position the effect of which will be eliminated by drilled holes around such counterbalances (at the correction planes, for instance) to values near zero, taking into account the effect of piston and rod weights (the later commonly replaced by equivalent "fictitious" weights). This intentional previous imbalance is designed just for the drilling phase, to take out mass (unbalance mass) from the external profile surface of the counterweight in a limited zone, restricted because of the engine geometry and manufacturing process. Figure 2-6 shows a schematic sketch of a crankshaft.

A crankshaft-block subsystem consists of the crankshaft and the engine block coupled by the main bearings. The cylinder pressure applied on the piston crown is transmitted to the crankpin through the piston-connecting rod assembly. The inertia of the piston-connecting rod provides a load on the crankpin as well. The loads of the crankpin deform the crankshaft and are transmitted to the engine block at the main bearing locations.

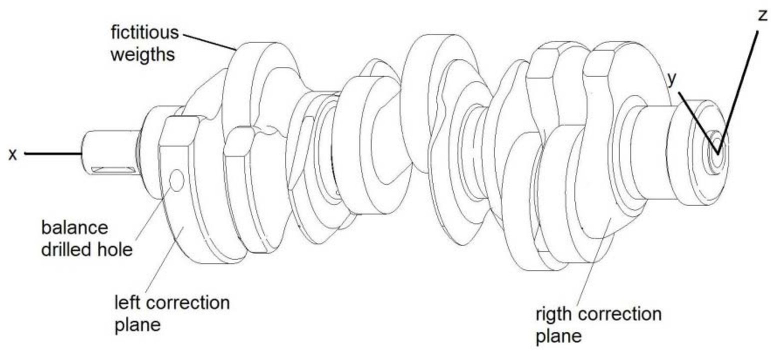



**Figure 2-6 Combustion Engine Crankshaft**

In general terms, imbalance is a condition in which the mass axis of a part does not coincide with the rotating axis. Imbalance is an expression of unequal distribution of mass around the rotating center (Imbalance =mass x radius). In a rotating part, unbalance causes vibration, which results in increased bearing loads, increased noise and vibration. Next, some terms and concepts are explained [61].

*Rotating axis*: Also known as shaft axis, rotating axis is a line about which a given body revolves. Two or more bearings usually determine it. In a crankshaft, all the mains define a cylinder that identifies it.

*Center of gravity*: The point where all the mass of a body can be considered concentrated. If hung from this point the body will be in equilibrium in all directions. Normally, any CAD software gives the mass properties of a part, including the location of the center of gravity related to any geometrical coordinate system.

*Mass axis*: It is an axis that goes through the center of gravity, about which a body will rotate if freely suspended, under no restriction. When the mass axis is not aligned to the axis of rotation, there is an imbalance. The imbalance can be created by attaching an external mass (balance weight) to the part. The same weight gives a different measure of unbalance if attached at different radii from the axis of rotation of the part. The effect of a balance weight increases as the radii (or correction diameter) increases.

*Static unbalance*: Static unbalance (also known as single plane unbalance) occurs when the mass axis is displaced parallel to the rotation axis. The amount and angle (location) of unbalance is the same at both ends of the shaft. As an explanatory example, if the part is allowed to rotate freely, it will rotate due to its own weight until the unbalance weight is at the bottom side. Static unbalance can be measured by either non-rotating equipment, or by rotating equipment, even with the mass properties if in CAD software, because it is manifested without part rotation. When rotating, static unbalance causes a radial motion described as a “hop”.

*Couple unbalance*: Couple unbalance arises when two equal unbalance masses are located at opposite ends of a rotating part, 180$^{o}$ degrees apart so, even when the resulting center of gravity is at the axis of rotation, the mass axis is not parallel to the rotating axis. Couple unbalance is only observable during part rotation, and is manifested in "two planes" see Figure 2-7. A crankshaft displaying a pure Couple Unbalance tends to "wobble".

*Dynamic unbalance*: Dynamic unbalance is a combination of both couple and static unbalance. This occurs commonly in parts that are at least twice as long as their diameter. To cause a dynamic unbalance, couple and static unbalance can be combined in different ways: a) The static and couple are in the same plane, and the geometric axis and mass axis intersect but the center of gravity is displaced; b) The unbalances are not equal and are opposite. The geometric axis and mass axis do not intersect. Dynamic unbalance can only be measured by rotating the part, and is manifested in two or more planes. It also causes the part also to wobble.

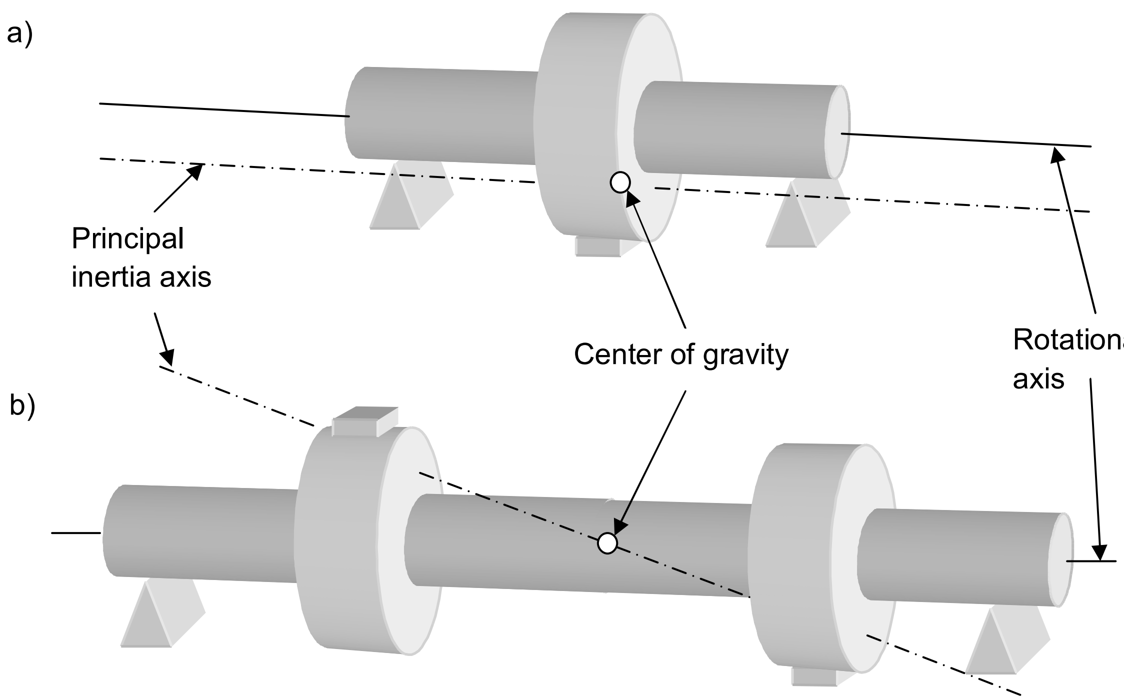



Figure 2-7 Types of unbalance: a) static, b) couple unbalance

When some parts are not symmetric by design, they are inherently imbalanced. Typical causes of imbalance in forging crankshafts are the distortion during stamping and cooling or handling fixture inaccuracies. During machining, causes are cutting tools wearing and tolerances. There are three correction methods to reduce unbalance: a) Add material (move mass center): welding, riveting, epoxing; b) Remove material (move mass center): Drilling, milling, punching, and nibbling; Mass center (move center of rotation).

### 2.2.2 Description of the balance equations

In a pivoted crankshaft turning with constant angular velocity, the sum of the moments produced by its mass differentials times their locations, is equal to the bending moment acting on the crankshaft. The bending moment is equal to the result of the moment caused by the centrifugal force due to the crankshaft mass at the centre of gravity, and the moment caused by the imbalance masses on the correction planes. Correction planes are located in both longitudinal sides of the crankshaft, where the external counterweights are located. These two counterweights are named CW1 and CW9, because correspond to the first and the ninth counterweigh. A design practice is to take the sum of moments around the left correction plane (into the counterweight close to the flange) to drop the moment caused by the left imbalance mass and find the imbalance on the right plane:

$$m_R r_{Ry} = \frac{\left(\frac{1}{2}\right) m_g\, r_{gy} + I_{xy}}{l} \qquad (1)$$

$$m_R r_{Rz} = \frac{\left(\frac{1}{2}\right) m_g\, r_{gz} + I_{xz}}{l} \qquad (2)$$

Where $l$ is the distance between the two correction planes, $m_g$ is the mass of the crankshaft; $r_{gy}$ and $r_{gz}$ are coordinates $y$ and $z$ of the centre of gravity and $I_{xy}$ and $I_{xz}$ the inertia products. In the same way, moments around the right correction plane are taken to find the imbalance on the left plane:

$$m_L r_{Ly} = \frac{\left(\frac{1}{2}\right) m_g\, r_{gy} - I_{xy}}{l} \qquad (3)$$

$$m_L r_{Lz} = \frac{\left(\frac{1}{2}\right) m_g\, r_{gz} - I_{xz}}{l} \qquad (4)$$

The mass properties required to calculate the imbalance ($m_g, r_{gy}, r_{gz}$ 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 imbalance as a response for fitness evaluation. 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 ($m_L r_L$ y $m_R r_R$) on the correction planes.

This method is called "goal programming", in which the designer has to assign targets or goals to be achieved for each objective function [62]. These values are incorporated into the problem as additional constraints. The algorithm will then try to decrease the absolute deviations from the targets to the objective functions. The simplest form of this method may be formulated as follows:

$$min \sum_{i=1}^{k} |f_i(x) - T_i| \; subject\ to\ x \in X \qquad (5)$$

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 by making modifications to the crankshaft geometry.

A convenient way to represent the imbalance of a crankshaft is by the use of correction polygons. Correction polygons are vector constructions that define the correction ability for a range of imbalances, of a set of holes (each represented by a vector) made upon the external side of some counterweights. Correction polygons can be created graphically: from the part print, determine the place and size of possible holes, each represented by a vector. From the geometric center of the crankshaft to the gravity center of the hole: $U = D_m\ x\ V_d\ x\ R_{cg}$, where $D_m$ represent the density of the material, $V_d$ is the volume of the hole and $R_{cg}$ is the radial distance from the center axis of the crankshaft to the center of the hole. When the vectors are drawn on polar paper, the length of the vectors represents the most unbalance that can be corrected by each hole at each vector angle. Ad vectors graphically from left and also from right. Any unbalance point inside the area covered can be reached by adding portions of vectors (that is, choosing the right number and size of holes).

The unbalance value of a crankshaft at each measuring plane can be plotted as a point showing amount and location. If a collection of values is plotted a balance scatter diagram can be constructed. The mean value of the unbalance can be used to correct the design to move the unbalance into the middle of the correction envelope.

As reference for the final results of the case study, Figure 2-8 shows the imbalance vector from the original crankshaft design to be optimized. It can be observed that the imbalance of both correction planes, even when inside the correction area, is not close to the target defined as 400 g-cm.

Balance is highly nonlinear with respect to the configuration of the crankshaft. The surfaces representing the performance metric being optimized can be complex. Gradient search techniques are prone to getting trapped in local minima. Thus, heuristic techniques such as genetic algorithms, which are all better at handling nonlinear objective spaces, need to be used, though these all require greater computational resources.

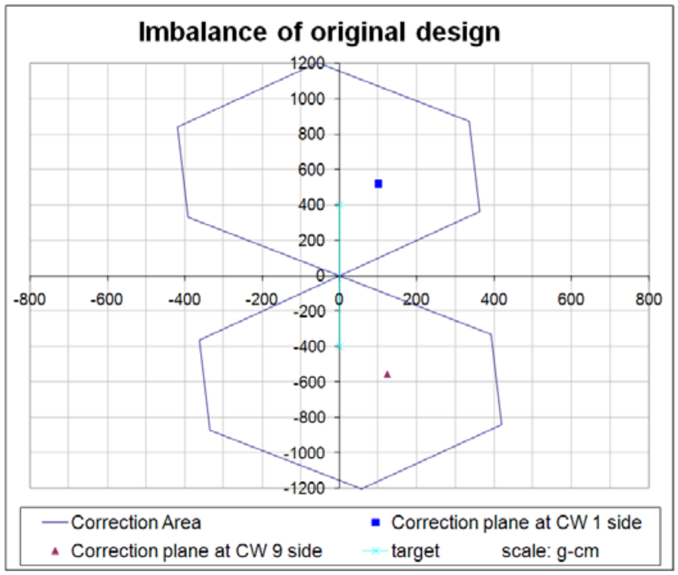



Figure 2-8 Imbalance graph from the original crankshaft design

## 2.3 Chapter conclusions

In this chapter the reasons for selecting a crankshaft as the case study are explained. The ever increasing requirements of an automotive engine aim for the use of forged steel in crankshafts. A good balancing design is a characteristic that has been required since the invention of the engine. It is explained that the goal of the design study is to reach the imbalance target and/or increase its dynamic response. All of this is performed in the context of a multi-objective problem. A general description of the structure of a crankshaft is presented. The development of the crankshaft according to the patents granted in the world, from the original patent from General Motors to the newer crankshafts, is chronologically described, including the only one which description is related to a forging.

The second section of the chapter relates to basic concepts of crankshaft dynamics. It is explained that the goal of the balance analysis of engine crankshaft in this research is to develop the geometry of the counterweights until the targets are reached. Traditionally, imbalance is corrected in the last step of the manufacturing via some drilling performed in the counterweight. Some definitions are provided and the different unbalance concepts are explained. Next an explanation of the imbalance functions and their derivation is developed. These equations are going to be introduced in the analysis of the CAD model in the next chapters. Finally, a graph in which the imbalance vector from the original crankshaft design can be seen is presented as reference for future improvements comparison.

# Chapter 3 Optimization Methods and Software

In this chapter is presented a bibliographic review of optimization methods and genetic algorithms. It is explained why a stochastic technique as the geometric algorithms is necessary for the kind of problems involved in the work. The multi-objective optimization strategies and why the case study was considered suitable for multi-objective analysis are also discussed.

## 3.1 Optimization Model

An optimization model is the base for every optimization and consists of variables, restrictions and targets within a system. In the system, all variables are defined by parameters, while during the optimizing process, the conditions of the restrictions can't be altered. The conditions of the restriction are set by the parameter´s limits. The optimization software chooses values for the variables from which these conditions are met. Among the conditions in the target system, the parameters are evaluated on their fitness. The target system is supported by a fitness function, which embodies what is either maximized or minimized, depending on the setting of the optimization software.

## 3.2 Optimization Methods

The choice of an optimization algorithm has to be made according to the problem. Figure 3-1 is a classification of different optimization methods taken from [63]. In the design process, one common problem is that they are typically nonlinear and, therefore, can be poorly optimized with analytical procedures. So an optimization algorithm for a nonlinear optimization and static parameters is needed. Genetic algorithms belong to a classification of global optimization algorithms. Gradient methods are suitable only when the solution to a problem implies the calculation of the partial derivatives and are used for local optimization. Under deterministic methods, the variation of parameters is used in certain logical strategies. They are characterized to progress in defined steps through the solution space in length and direction. In the stochastic process, the parameters are chosen purely at random and are not logically moved by a predetermined strategy. A special case of stochastic methods, the evolutionary algorithms, can generate completely new solution variants.

Parametric optimization using the different algorithms is perhaps the most effective approach for many industrial solutions, as commonly parametric changes in products and process are easier to achieve and to implement than innovative concepts, where shape, topology, or physical principles are changed. However, parametric optimization alone could lead to stagnation in product or process development, as compromise is inherent in parametric optimization, especially when multiple optimization objectives are targeted. Multi-objective optimization requires definition of "priorities" from among conflicting performance objectives, therefore leading to compromises in conflicting goals. These conflicting performance goals appear in any product or process development process, and consequently achievement of further enhancements through parametric multi-objective optimization.

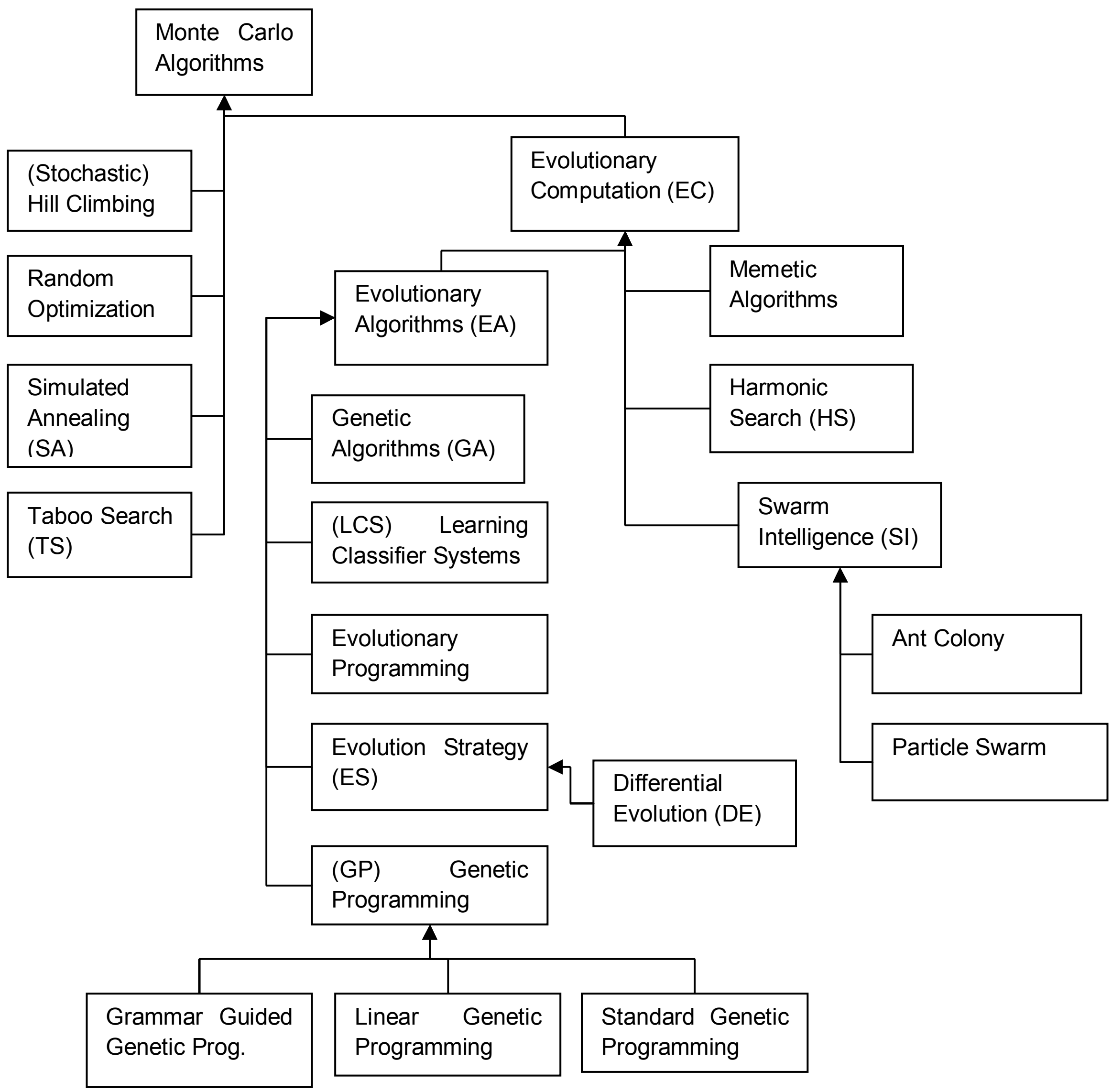



**Figure 3-1 Probabilistic global optimization algorithms**

Products and processes optimization has been studied by many authors, especially since the widespread use of computers as an aid for looking for an "optimal" combination of product and process parameters. The introduction of new techniques for design of experiments (DOE) in product or process improvement made it possible to reduce the number of experiments needed to identify the influence of different parameters in the performance objectives. DOE also facilitated obtaining empirical mathematical models of the products and/or processes leading to the application of multi-objective optimization methods. Furthermore, evolutionary and genetic algorithms in engineering optimization have contributed to the achievement of higher performance goals with multi-objective optimization. Nevertheless, these techniques had been restricted to the search for product or process performance enhancement through the variation of numerical product or process parameters.

One of the first applications of genetic algorithms was the learning classifier system (LCS) developed by Holland. Even when it is not strictly an optimization algorithm, it consisted of a population of binary rules on which a genetic algorithm altered and selected the best rules. A complete bibliography of books related to LCS can be found in [64]. An original patent describing of the use of genetic algorithms in computer systems was granted to Holland in 1987 [65]. The invention relates to electronic data processing systems and, more specifically, to adaptive, parallel, rule-based computing systems called "classifier systems". Classifier systems adapt to the computing task by selecting useful processes from a collection of available alternatives, and by further experimentation with new candidate processes, which are added to the pool of alternatives in a quasi-random manner. An adaptive classifier system selects and further improves the computational processes it uses in much the same way that nature adapts biological species to their environment: by producing new candidates, nurturing the successful, and discarding those which perform poorly.

The patent explains that, as in biological evolution, means are employed both to insure the survival of those processes exhibiting superior performance, and to form new processes which are composed of elements copied from pairs of successful but differing parent processes. Whenever a classifier generates a message which is carried over into the next major cycle, it rewards the classifier(s) which supplied the input message by increasing the strength of the supplying classifiers (which share the reward equally) and decreasing the strength of the supplied classifier (which will itself be rewarded if its message is used in a subsequent cycle). In this way, the strength of each classifier producing useful messages is enhanced over time, while the strength of those that are unable to produce useful messages deteriorates.

According to another feature of the invention, the number of messages passed on to a subsequent cycle is limited to a number less than the total number of messages that would be generated by classifiers whose condition(s) are satisfied. Thus, means are employed for discarding messages generated by weaker and more general classifiers in favor of messages produced by classifiers having greater associated strength values and which are more specific (that is, which respond to a more limited number of messages). Pairs of classifiers having high relative strength values are employed to form new combination classifiers using random genetic operators, the most important of which, called "crossover", involves the selection of a string position at random, splitting both parent classifiers at that position, and exchanging parts to form two new offspring classifiers which replace the weakest (lowest strength) classifiers.

From the classifier systems, Holland derived that the genetic algorithms are a procedure for increasing the adaptive scope of classifier systems. These algorithms periodically select the best or fittest classifiers, produce new classifiers from them, and substitute these new classifiers for some of the poorest classifiers.

## 3.3 Use of Genetic Algorithms

Genetic algorithms, an evolutionary computational tool, are selected to be integrated as part of the strategy. Genetic algorithms (GAs) are adaptive heuristic search algorithms (stochastic search techniques) based on the ideas of evolutionary natural selection and genetics [66]. The basic concept of GAs is aimed at simulating processes in natural systems that are necessary for the mechanics of evolution, specifically, those processes that follow the principles proposed by Charles Darwin: the “law of the strongest” or the “survival of the fittest”.

They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) or individuals is created using the fittest as parents. While randomized, GAs are no simple random stroll. They efficiently exploit historical information to speculate on new search points with expected improved performance [66].

### 3.3.1 Darwin’s theory of natural selection in brief

Darwin’s theory of natural selection [67] has had the most important consequences for our perception of change. His view of evolution refers to a continuous, gradual adaptation of species to changes in the environment. The idea of ‘survival of the fittest’ means that the most adaptive organisms in a population will survive. This occurs through a process of ‘natural selection’ in which the most adaptive ‘species’ (organisms) will survive. The process is gradual, taking place in a relatively stable environment, working slowly over long periods of time necessary for the distinctive characteristics of species to show their superiority in the ‘survival contest’. This year (2009) is the 200th anniversary of Darwin’s birth and his ideas are crossing frontiers into the computer world. This research thesis is a small contribution dedicated to his memory.

Three basic concepts derived from Darwin’s evolutionary biology are what the computer geneticists are trying to extend to the algorithms [68]. These three concepts are the unit of variation, unit of selection and unit of evolution.

The unit of variation concerns the entity which contains the genetic information and which mutates following specific rules, namely the genes. Genes contain the hereditary information which is preserved in the DNA. This is not altered throughout the reproductive lifetime of an organism. Genes are passed on from an organism to its successors. The gene pool, i.e., the total stock of genetic structures of a species, only changes in the reproduction process as individuals die and are born. Particular genes contribute to distinctive characteristics and behavior of species which are more or less conducive to survival. The gene pool constitutes the mechanism to transmit the characteristics of surviving organisms from one generation to the next.

The unit of selection is the expression of those genes in the entities which live and die as individual specimens, namely (individual) organisms. These organisms, in turn, are subjected to a process of natural selection in the environment. 'Fit' organisms endowed with a relatively 'successful' gene pool, are more likely to pass it on to their progeny. As genes contain information to form and program the organisms, it can be expected that in a stable environment genes aiding survival will tend to become more prominent in succeeding generations.

Finally, there is the unit of evolution, the concept that refers to changes over time as the gene pool changes, namely populations. Natural selection produces changes at population level by 'trimming' the set of genetic structures in a population. 'Natural selection', thus, is a gradual process selecting the 'fittest' organisms.

### 3.3.2 Introduction to Genetic Algorithms

Genetic algorithms are typically used in computer simulation, normally used for function optimization, in which a population of abstract representations (called chromosomes) of possible solutions (called individuals) A gene is defined as any portion of chromosomal material that potentially lasts for enough generations to serve as a unit of natural selection [1]. Traditionally, the solutions are represented in binary code as strings of 0s and 1s, but other codes are also possible. The population evolves in discrete generations. At the beginning of each generation, the computer starts by constructing a phenotype for each individual, using the set of number strings corresponding to an individual (the "genome") as a specification. The function used for the interpretation is dependent upon the experiment, but typically a fixed region within the chromosomes is used to determine each phenotypic trait of the individual. The following explanation of optimization with an evolutionary algorithm is shown in Figure 3-2.

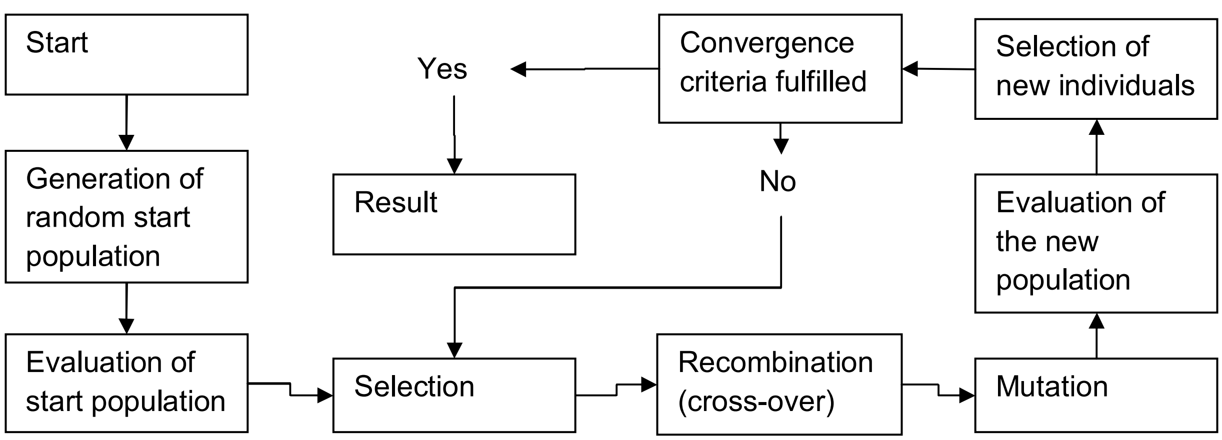



**Figure 3-2 Standard loop of genetic algorithms**

A GA is an iterative procedure, which maintains a constant-size population $P(t)$ of candidate solutions, each individual in the population represents a point in the search space of the problem to be solved. During each iteration step, called a generation, the structures in the current population are evaluated, and, because of those evaluations, a new population of candidate solutions is formed. The initial population $P(O)$ can be chosen heuristically or at random. The structures of the population $P(t+1)$ are chosen from $P(t)$ by a randomized selection procedure that ensures that the expected number of times a structure is chosen is approximately proportional to that structure's performance relative to the rest of the population.To simulate selection, the phenotypes are scored according to a set of fitness criteria. When the system is being used to solve an optimization problem, the traits are interpreted as solution parameters and the individuals are scored according to the function being optimized. This score is then used to cull the population in a way that gives higher scoring individuals a greater chance of survival. The aptitude of an individual is closely related to the value of the function in the point being represented by the individual. After the selection step, the surviving gene pool is used to produce the next generation by a process analogous to mating. Mating pairs are selected by either random mating from the entire population, some form of inbred mating, or assertive mating in which individuals with similar traits are more likely to mate. The pairs are used to produce genetic material for the next generation by a process analogous to sexual reproduction.

In a simple GA, the whole population is replaced by a new set of individuals each generation. The new set of individuals is produced in pairs. In order to produce two new individuals, a pair of parents is selected from the current population. Those individuals with a better aptitude have more chances of being selected. Once a pair of individuals is selected, crossover and mutation are applied. The crossover consists of constructing a pair of new individuals by taking parts of the genetic material from both parents. The expected effect is the combination of the characteristics present in both parents. In the simplest case, the genetic material of an individual consists of the string and the crossover consists of randomly taking a point in which both parents can simultaneously be divided and then joining the first part of the first parent with the second part of the second parent. The second individual can be constructed with the remaining parts of the parents' genetic material.

The chance coming together, through crossing-over, of previously existing sub-units is the usual way for a new genetic unit to be formed. Another way—of great evolutionary importance even though it is rare—is called point mutation. A point mutation is an error corresponding to a single misprinted letter in a book. It is rare, but clearly the longer a genetic unit, the more likely it is to be altered by a mutation somewhere along its length. Another rare kind of mistake or mutation which has important long-term consequences is called inversion. A piece of chromosome detaches itself at both ends, turns head over heels, and reattaches itself in the inverted position. In terms of the earlier analogy, this would require some renumbering of pages. Sometimes portions of chromosomes do not simply invert, but become reattached in a completely different part of the chromosome, or even join with a different chromosome altogether.

However, like other classes of algorithms, GAs differ from one another with respect to several parameters and strategies. The following are the main concepts related to genetic algorithms that are commonly used:

*Population Size (*$\mathrm{N}$): The population size affects both the ultimate performance and the efficiency of GAs. GAs generally do poorly with very small populations, because the population provides an insufficient sample size for most representations.

*Crossover Rate (*$C$): The crossover rate controls the frequency with which the crossover operator is applied. In each new population, $C * \mathrm{N}$ structures undergo crossover. The higher the crossover rate, the more quickly new structures are introduced into the population.

*Mutation Rate ($M$):* Mutation is a secondary search operator which increases the variability of the population. After selection, each bit position of each structure in the new population undergoes a random change with a probability equal to the mutation rate $M$.

*Generation Gap ($G$):* The generation gap controls the percentage of the population to be replaced during each generation. That is $N * (G)$ structures of $P(t)$ are chosen (at random) to survive intact in $P(t + 1)$.

*Scaling Window* ($W$): When a numerical function $f(x)$ is being maximized with a GA, it is common to define the performance value $u(x)$ of a structure $x$ as $u(x) = f(x) - f_{min}$, where $f_{min}$ is the minimum value that $f(x)$ can assume in the given search space.

*Selection Strategy* ($S$): A good strategy assures that the structure with the best performance always survives intact into the next generation. In the absence of such a strategy, it is possible for the best structure to disappear, due to sampling error, crossover, or mutation.

There are some fundamental differences between evolutionary algorithms and other deterministic or stochastic optimization procedures:

a) The search starts from several points simultaneously, not just one.
b) There are no discharges of objective function.
c) There are probability rules, not deterministic rules.
d) There is a range of possible solutions.
e) The user makes the final selection of the best draft by decision.
f) Continuous and discrete variables can be used.
g) A choice can be made between continuous, discontinuous and multi-function space.
h) They are easy and flexible to apply, since the objective function is not subject to restrictions
i) They have the advantage of producing many configurations in a single run, giving a "Pareto front" of many optimal solutions at once.

Genetic algorithms are a population-based technique. This means that they will produce many optimized configurations simultaneously, which is very beneficial for multi-objective optimization where a Pareto front of solutions is desired. They also allow many different initial guesses for the configurations to be considered at once [33]. This cycle continues until a certain criterion is reached. Experimental studies indicate that GAs exhibit extremely high efficiency, consistently outperforming both gradient techniques and various forms of random search [69]. Eventually, the population converges to a single solution which is likely the best found and hopefully the best that could ever be found. A GA's stochastic operators allow the algorithm to search intractable spaces with some hope of finding optimal solutions within a computationally reasonable time frame. The GA cycle can be summarized as follows:

a) [**Start**] Generate random population of $n$ chromosomes (suitable solutions for the problem)
b) [**Fitness**] Evaluate the fitness $f(x)$ of each chromosome $x$ in the population.
c) [**New population**] Create a new population by repeating following steps:
   i. [**Selection**] Select two parent chromosomes from a population according to their fitness (the better the fitness, the greater the chance to be selected).

ii. [**Crossover**] With a certain probability, cross the parents over to form new offspring (children). If no crossover was performed, offspring are an exact copy of the parents.
iii. [**Mutation**] With a certain probability, mutate new offspring at each position in the chromosome.
iv. [**Accepting**] Place new offspring in a new population.

d) [**Replace**] Use the new generated population for another run of the algorithm.
e) [**Test**] If the end condition is satisfied, stop and return to the best solution so far.
f) [**Loop**] Return to b)

Some of the conditions preferable for a good evolutionary development include:

a) A good genetic representation of the individuals
b) A varied initial population
c) Selection of good parents for crossover to produce good offspring
d) Selection of the best offspring to be parents in the next generation
e) Correct refinement of the genetic algorithm.

Next is a detailed description of terms that are part of the vocabulary of genetic algorithms.

### 3.3.3 Start population

The startup population is a set of individuals usually generated at random. However, it is also possible to start a population from an existing or a predefined file. The boundaries of the generation of individuals are defined by the restriction system. The quality of a startup population can be positively affected by seeding “good" individuals into it. However, there is still the risk of achieving a premature convergence and not finding the global optimum. From a startup population randomly generated, every optimization gives different results, whereas at a specified optimization seed, different algorithms produce exactly the same result.

### 3.3.4 Fitness function

The fitness function corresponds to the objective function used in the optimization and is crucial for calculating the probability of survival. The optimization software, depending on its configuration, can be used for minimizing or maximizing.

### 3.3.5 Selection

The selection procedure used also contributes to the optimization solution. The selection of individuals for the generation of offspring is what delivers the solution. Many methods have been developed. A choice can be made between static and dynamic selection: in dynamic selection the probability of selection in each generation is calculated and in static selection only the first generation will be assessed for future generations. A distinction can also be made between discriminatory and non-discriminatory selection. In discriminatory selection, poor individuals are deleted, making the differences in quality of the remaining individuals much stronger. When an individual is rated as poor, it must be reported to the algorithm. Non-discriminatory procedures have the advantage of applying a thorough search in the solution space before a promising area is chosen. The most common procedures for selection are presented next.

#### *3.3.5.1 Roulette selection*

Each individual section is a slide in a circle on a roulette wheel allocated in proportion to its fitness. A random number draws the selection of individuals. The random number is the turning of the roulette wheel. The individual referred to by the pointer is then being chosen. The probability of being selected is particularly high in a larger section on the roulette wheel. There are two different manifestations of the method. In the normal roulette wheel selection, the individuals remain, even after the selection on the roulette wheel and can thus be selected again. In the unique roulette wheel selection, the individual is removed from the roulette wheel and the sections are recalculated. This process is repeated as long as individuals are required.

#### *3.3.5.2 Roulette wheel with multiple selection.*

This procedure is a modified roulette wheel selection. Around the roulette wheel there are so many pointer distributed, with the individuals that should be selected evenly positioned. The allocation of the section is proportional to the fitness of the individuals. The even distribution and the proportional allocation of the sections in this process ensures that the best individual selection survives in any case.

#### *3.3.5.3 Linear Ranking*

This procedure will try to get a convergence against a premature suboptimum in order to prevent it. The individuals will be arranged and ranked on the basis of their fitness levels, starting with one value assigned. The selection will have a probability of rank-dependent linear function. Here, poor individuals get a higher chance of selection, and the dominance of very good individuals will be reduced. The idea is straightforward. Sort the population from best to worst, assign the number of copies that each individual should receive according to a non-increasing assignment function, and then perform proportionate selection according to that assignment [70].

#### *3.3.5.4 Tournament selection*

The tournament idea is simple. Choose some number of individuals randomly from a population (with or without replacement), select the best individual from this group for further genetic processing, and repeat as often as desired (usually until the mating pool is filled). Tournaments are often held between pairs of individuals (tournament size s = 2), although larger tournaments can be used and may be analyzed.

### 3.3.6 Recombination

Recombination is the creation of new individuals from parts of several existing individuals by the crossing (crossover) of chromosomes. It will be carried out in accordance with the selection process: individuals selected from the population and crossed with each other. This is repeated until the percentage of crossings carried out is reached. In an odd number of crossings the last individual receives no opportunity to cross. The main procedures are as follows:

#### *3.3.6.1 Two-point crossover*

In a two-point recombination, two parts are exchanged between individuals and there are two new individuals. The crossing points are two random numbers and, therefore, can vary at each recombination.

#### 3.3.6.2 Multi-point crossover

The recombination is regarded as analogous to the two-point crossover, but with more than two crossing points.

#### 3.3.6.3 Parametric uniform crossover

The recombination in this process takes place by the crossing points. For each parameter a separate decision is made as to whether it will be exchanged or not. The exchange is likely to be in general between 50% and 80%. The special case with the exchange probability of 50% is called uniform crossover.

#### 3.3.6.4 Random crossover

The purpose of this procedure lies in the elimination of the dependence on the parameters during recombination. This is achieved through a random substitution of the parameters before the recombination. After applying the operator, the parameters return to the original position. The actual operation in this procedure is a single point crossover, i.e. recombination using a single crossing point.

#### 3.3.6.5 Linear recombination

In this process, the parameter values of two individuals are represented by imaginary lines. In the normal line recombination, these parameter values are in the middle of the line; in the advanced recombination the parameters can be anywhere in these connection lines

Another concept related to Crossover is Hybridization, which acts as a combination at the phenotypical level and is a strong and useful concept in the context of evolutionary computation. Hybridization for individuals of any species depends a great deal upon the characteristics of each species. In the case of geometric design, it is necessary to think in term of the basic geometries that define each of the features that compose a 3D solid model within a CAD program. These geometries must be interpreted as generic individual representing each geometric feature of the real model. An abstract representation allows a genetic codification capable of taking part in crossover and mutation processes. The result of the alteration of the genetic material by crossover or mutation, must then be reinterpreted and converted back into a phenotypic geometric form. This produces a new geometry with different features and characteristics that must be evaluated. Its performance under given conditions determines its fitness and its probability of surviving as a better design.

### 3.3.7 Mutation

Some types of mutation include:

#### 3.3.7.1 Adaptive mutation

In adaptive mutation is defined the adaptation of the mutation step in the quality of the generation and the progress of the optimization is defined. During the optimization the mutation step is reduced to local for better results. If the algorithm only finds a local optimum, leading to stagnation of the optimization, it is possibly because the mutation step is not enough to bring it to reach optimum

#### 3.3.7.2 Sense of mutation

In evolutionary algorithms there is often the problem that they can converge in a local optimum. This occurs when individuals early and randomly are enforced to reach the local optima. This increases the likelihood that they increasingly dominate the population and, in extreme cases, at the end all individuals in a population are identical. In such a case, new individuals can no longer be generated by crossings, as the existing individuals are identical. Here the mutation helps.

There is the notion that a mutation in a particularly small population is beneficial. In the case of a large population, it is precisely the opposite, because there is a much greater diversity in the population and due to mutations, good parameter values can be destroyed. In the case of rapidly changing fitness values, the mutation is also important, as new variants quickly appear in the population.

### 3.3.8 Replacement strategy

After the execution of the genetic operations, there must be a transition from the old generation to the newly created individuals that formed a new generation. There are different strategies for this.

#### 3.3.8.1 Random replacement

Individuals are randomly select until a new generation is created. This can be done by choosing how many individuals from the old generation will be taken over. The remaining seats in the new generation will be newly drawn individuals. In this method, there is a high risk of losing good individuals.

#### 3.3.8.2 Elitist replacement

Here, the best of the individuals (number one) of the current generation is chosen, and the remaining individuals are added randomly from the newly created individuals. This also has the risk of losing good solutions.

#### 3.3.8.3 Roulette wheel replacement

Each individual represented on a section of a roulette wheel is allocated in proportion to its fitness. The selection of individuals is drawn by a random number; the larger the section on the roulette wheel, the higher the probability to be selected. After the selection is made, an individual from the roulette wheel is removed and the probabilities for the remaining individuals are recalculated. This process is repeated until enough individuals for the new generation have been selected.

### 3.3.9 Termination criteria

Ending the optimization can be based on different criteria. The most important refer to the following:

a) Reaching a maximum number of calculations
b) Reaching a maximum number of generations
c) Finding a solution with a specified minimum
d) In excess of a given medium improvement in the last few generations

### 3.3.10 Genetic algorithms in simulation

Genetic algorithms have been studied extensively and have been applied in many fields of engineering. Many real-world problems involving a search for optimal parameters may be hard to solve with traditional methods, but when GAs are used the solution is more easily found. However, due to their outstanding performance in optimization, GAs have been wrongly regarded as a tool only for optimization. Genetic algorithms also show impressive results compared to other search engines, and for this reason the authors perceive them as a potential tool for creativity enhancement. Genetic algorithms, by being global optimization techniques, avoid many of the shortcomings exhibited by local search techniques on difficult search spaces [66].

The power of GAs derives largely from their ability to exploit efficiently this vast amount of accumulating knowledge by relatively simple selection mechanisms. Termination of the GA may be triggered by finding an acceptable approximate solution, by fixing the total number of structure evaluations, or some other application dependent criterion. In addition, a number of experimental studies show that GAs exhibit impressive efficiency in practice [71]. While classical gradient search techniques are more efficient for problems which satisfy tight constraints, GAs consistently outperform both gradient techniques and various forms of random search on more difficult (and more common) problems, such as optimizations involving discontinuous, noisy, high-dimensional, and multimodal objective functions.

Recent studies have developed approaches close to the research line that this work is pursuing. For example, a patent from 2009 [72] is related to a computer-aided design method, and more specifically, to designing concept sketches, such as for automobile profiles, using a genetic algorithm. The patent explains the problems confronted with these kinds of approaches:

“The conventional design process which involves a "proof of concept" has been accomplished with drawings and subsequent refinements done by hand or with computer assistance. A designer creates concept sketches or profiles, such as for a new automobile, by hand or with a suitable CAD/CAM program. The process may take up to one or two months, or longer, to create acceptable concepts. An average designer may create about twelve sketches a day. Renderings in one-fifth scale may be achieved in less than two days.

“Such conventional techniques are time consuming and tedious. In addition, traditional CAD/CAM programs are, for the most part, enhanced drawing tools which merely reflect the designer's creativity but do not generate creative contributions. The conventional approach does not dramatically reduce the cycle time to generate acceptable proofs of concept, such as for a new automobile”

The patent claims that an objective of the invention is to shorten design development time and allow a designer to quickly generate high quality concepts or profiles. Another objective is to increase design productivity and quality with a program that learns the designer's preferences and incorporates the designer's preferences into the generation of new offspring profiles. These objectives are achieved by generating a family tree identifying successive generations of the parent and offspring profiles.

### 3.3.11 Example of a topology optimization

The next example is presented to show how a simple mechanical problem can benefit from the genetic algorithms and the ever increasing capacity of computers to perform FEM analysis in a short time.

A bridge is submitted for analysis and weight optimization (see Figure 3-3) with the following design restrictions:

a) The bridge can not bend more than current design
b) Internal stress of elements cannot be higher than 30, the current yield point

The Genetic Algorithm used was vgGA [73]. The virtual gene genetic algorithm (vgGA) is a generalization of traditional genetic algorithms that use binary linear chromosomes. This implementation allows the generalization to virtual chromosomes of alphabets of any cardinality. VgGa is an algorithm that implements crossover and mutation as arithmetic functions of the phenotype of the individuals.

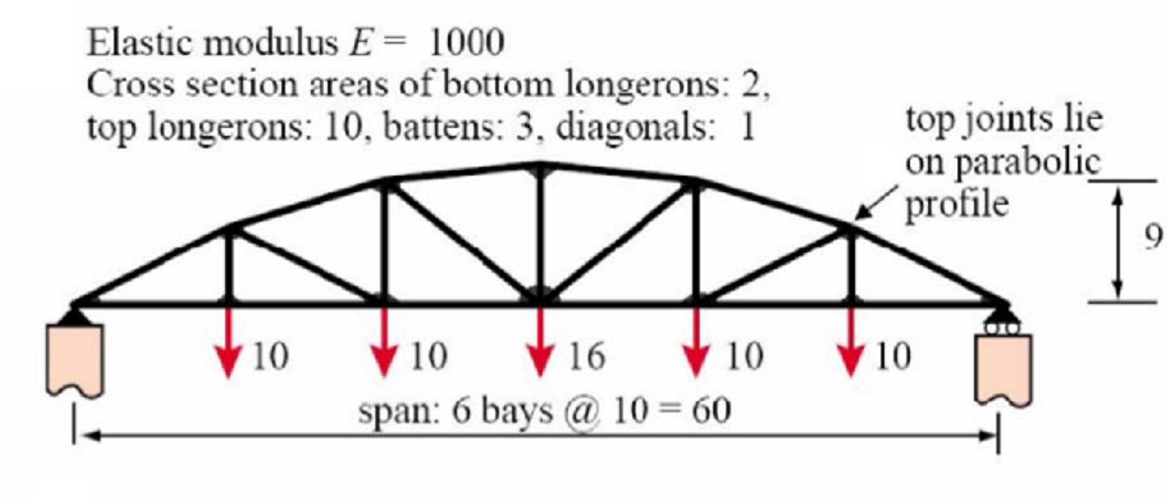



Figure 3-3 Example bridge characteristics.

The attempt considered only bending restriction. Ranges of each area section the GA are limited as follows:

Bottom longerons= .5-2.5 $m^2$

Top longerons= 5-11 $m^2$

Battens: 1-3.5 $m^2$

Diagonals: .2-1.2 $m^2$

The genetic algorithm used was vgGA

Mutation probability: $P_m = .01$

Crossover probability: $P_c = .9$

individuals $(n)$ /generation: $n = 300$

generations $(g)$: $g = 200$

Objective function: $f(x) = 1/w$ where $w$ is the weight of bridge structure

The objective is to minimize Weight given a deformation limit $(umax)$, that is, max displacement of each node to be less than a fixed limit. The program performs a "sudden death" to individuals trespassing the bending limit and assigns a value of $f(x) = 0$.

$$umax < 2.55$$

$$Minimize\ W$$

The Figure 3-4 shows the weight of the bridge as the algorithm evolves. An increase in the weight of the population at the beginning of the evolution is followed by an asymptotic reduction.

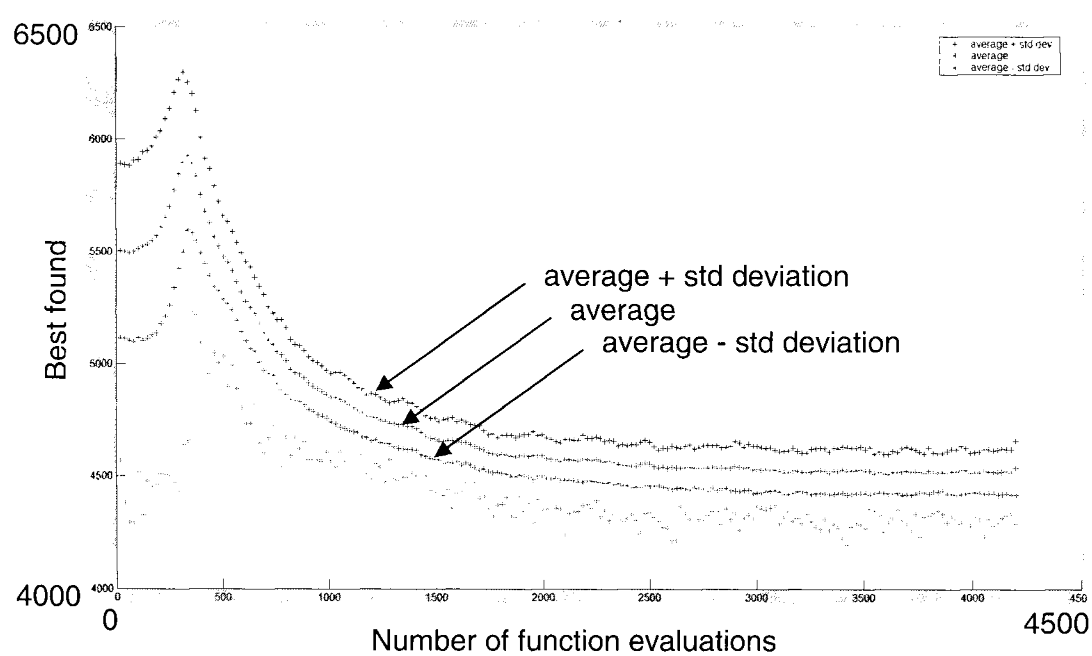



Figure 3-4 Evolution of bridge optimization

The optimal Area for each longeron is given:

Bottom longerons: 2.5,2.5,2.5,2.5,2.5,2.5

Top longerons: 5,5.94,5.94,5.75,5.56,5.18

Battens: 1,1,1.15,1,1

Diagonals: 0.356,0.73,0.79,0.42

Total Weight (not considering yield stress restriction): =4422.5kg (vs original 7124.2Kg it is a reduction of 38%)

The example gives the general description of the proposed method for application of the GAs to mechanical design and a promising example of such an application. The results with the bridge demonstrate an interesting way to design based on optimization using GAs.

## 3.4 Multi-objective Optimization

Evolutionary and genetic algorithms in engineering optimization have contributed to the achievement of higher performance goals with multi-objective optimization. Nevertheless, these techniques have been restricted to the search for product or process performance enhancement through the variation of numerical product or process parameters. Parametric optimization is perhaps the most effective approach for many industrial solutions, as commonly parametric changes in products and process are easier to achieve and to implement than innovative concepts, where shape, topology or physical principles are changed. However, parametric optimization alone could lead to stagnation in product or process development as compromise is inherent in parametric optimization; especially when multiple optimization objectives are targeted. An exhaustive survey of Multi-objective optimization techniques can be found in [32]. A state-of-the art analysis of multi-objective evolutionary algorithms is found in [74].

Genetic algorithms, are well suited to searching intractably large, poorly understood problem spaces, but have mostly been used to optimize a single objective. They all describe a scalar value to be maximized or minimized. However, a careful look at many, if not most, of the real-world GA applications reveals that the objective functions are really multi-attribute. Many optimization problems have multiple objectives. Historically, multiple objectives have been combined ad hoc to form a scalar objective function, usually through a linear combination (weighted sum) of the multiple attributes, or by turning objectives into constraints. Often-seen tools for combining multiple attributes are constraints, with associated thresholds and penalty functions, and weights for linear combinations of attribute values. Some authors propose performing a set of mono-objective optimization tasks to reveal conflicts [75], which would give the values of all the objective functions an “acceptable trade off” to the designer [62].

The following two features are desired to solve multi-objective problems successfully: 1) the solutions obtained are Pareto-optimal and 2) they are uniformly sampled from the Pareto-optimal set. John Koza [76], leader in genetic programming, pointed out: “Representation is a key issue in genetic algorithm work because genetic algorithms directly manipulate the coded representation of the problem and because the representation scheme can severely limit the window by which the system observes its world”. Coello has developed a multi-objective optimization tool called MOSES (Multi-objective Optimization of Systems in the Engineering Sciences) [77]. The results produced by MOSES are compared to those produced with other mathematical programming techniques and GA-based approaches and show the new techniques' capability to generate better trade-offs than the approaches previously reported in the literature.

An analysis of multi-objective optimization with genetic algorithms can be seen in [78]. In a typical multi-objective optimization problem, there is a set of solutions that are superior to the rest of the solutions in the search space when all objectives are considered but are inferior to other solutions in the space in one or more objectives. Designers are commonly interested in a set of Pareto-optimal points when solving multi-objective problems. Since generic algorithms work with a population of points, it seems natural to use GAs in multi-objective optimization problems to capture a number of solutions simultaneously. One solution chosen by a designer may not be acceptable to another designer or in a changed environment. Therefore, in multi-objective optimization problems, it may be useful to have knowledge about alternative Pareto-optimal solutions [79]. Studies on multi-objective GAs started with the pioneering work by Schaffer in 1984 [52] and, since then, there has been a number of different types of multi-objective genetic algorithms. A fundamental motivation behind these studies is the population-based search ability of GAs in which a population of individuals captures multiple Pareto optimal solutions in a single run. Moreover, if available, a decision maker may be interested in knowing alternate solutions. As genetic algorithms work with a population of points, a number of Pareto-optimal solutions may be captured using GAs.

A general multi-objective optimization problem consists of a number of objectives and is associated with a number of inequality and equality constraints. Mathematically, the problem can be written as follows:

Find

$$x$$

In order to Minimize/Maximize

$$f_i(x) \mid i = 1,2,\ldots,N$$

subjected to

$$g_j(x) = 0;\ j = 1,2,\ldots,J$$
$$h_k(x) = 0;\ k = 1,2,\ldots,K$$

The parameter $x$ is a $p$ dimensional vector having $p$ design or decision variables. A few studies have tried a different approach to multi-criteria optimization with GAs: using the GA to find all possible trade-offs among the multiple, conflicting objectives. These solutions (trade-offs) are non-dominated, in that there are no other solutions superior in all attributes. In attribute space, the set of non-dominated solutions lies on a surface known as the Pareto optimal frontier [80]. The goal of a Pareto GA is to find and maintain a representative sampling of solutions on the Pareto front. Solutions to a multi-objective optimization problem are mathematically expressed in terms of non-dominated or superior points. In a minimization problem, a vector $x(1)$ is partially less than another vector $x(2)$;($x(1) \leq x(2)$),when no value of $x(2)$ is less than $x(1)$and at least one value of $x(2)$ is strictly greater than $x(1)$. If $x(1)$ is partially less than $x(2)$, one says that the solution $x(1)$ dominates $x(2)$ or the solution $x(2)$ is inferior to $x(1)$. Any member of such vectors that is not dominated by any other member is said to be non-dominated or non-inferior. Similarly, if the objective is to maximize a function a dominated point is defined, if the corresponding component is not greater than that of a non-dominated point. The optimal solutions to a multi-objective optimization problem are non-dominated solutions. They are also known as Pareto-optimal solutions. Mathematically, an optimization algorithm should be terminated if any one of the Pareto-optimal solutions is obtained. However, in practice, since there could be a number of Pareto-optimal solutions and the suitability of one solution depends on a number of factors including designer's choice and problem environment, finding the entire set of Pareto-optimal solutions may be desired. Next, a description of a number of classical approaches to the solution of multi-objective optimization problems will be discussed.

A common difficulty with multi-objective optimization is the appearance of an objective conflict. None of the feasible solutions allows simultaneous optimal solutions for all objectives. In other words, individual optimal solutions for each objective are usually different [81]. Thus, a mathematically most favorable Pareto-optimum is that which offers the least objective conflict. Such solutions can be viewed as points in the search space that are optimally placed from the individual optimum of each objective. However, such solutions may not satisfy a decision maker because he or she may want a solution that satisfies some associated priorities of the objectives. To find such points all classical methods scalarize the objective vector into one objective. As mentioned, many classical algorithms for nonlinear vector optimization techniques define a substitute problem, reducing the vector optimization to a scalar optimization problem. Using such a substitute, a designer can find a compromise solution, subjected to specified constraints.

However, multi objective optimization requires that "priorities" be defined among conflicting performance objectives, therefore leading to compromises in conflicting goals. Nevertheless, as in some cases conflicting performance goals may be achieved through parametric multi-objective optimization, innovative changes such as shape or topological variations, or such as the change of physical principles, perhaps may not be strictly necessary. Conversely, in many cases shape or topological variations or changes in physical principles often lead to better solutions that overcome conflicting performance goals (technical contradictions). The next three methods are the most commonly used: method of objective weighting, method of distance functions, and method of min-max formulation.

*Method of Objective Weighting.* This is probably the simplest of all classical techniques. Multiple-objective functions are combined into one overall objective function.

*Method of Distance Functions.* In this method, the scalarization is achieved by using a demand-level vector $y$ which has to be specified by the decision maker. Here, a single objective function is derived from multiple objectives.

*Min-Max Formulation.* This method is different in principle from the above two methods. It attempts to minimize the relative deviations of the single objective functions from the individual optimum. That is, it tries to minimize the objective conflict.

The most profound drawback of these algorithms is their sensitivity toward weights or demand levels. The decision maker must have a thorough knowledge of the priority of each objective before forming the single objective from a set of objectives. The solutions obtained largely depend on the underlying weight vector or demand level.

## 3.5 Optimization Package: DAKOTA

The optimization used in this method is made with the program package DAKOTA (Design Analysis Kit for Optimization and Tera Scale Applications) [82]. Developed at Sandia Laboratories, DAKOTA is an optimization framework with the original goal of providing a common set of optimization algorithms for engineers who need to solve structural and design problems. DAKOTA offers a flexible and extensible interface between analysis programs and optimization algorithms. In DAKOTA there are many different algorithms, with emphasis on non-gradient based optimization procedures. Since this package is a central component of the method, the following is an overview of DAKOTA. The DAKOTA toolkit is designed to help with technical problems and to offer support for product design optimization. Particularly in the area of structural mechanics, heat transfer and flow simulation, a better understanding of the complex physical connection is possible. It can be used for a reduction in weight; cost and the probability of failure of products.

### 3.5.1 How DAKOTA functions

There is a close binding between the simulation and DAKOTA. The data is loaded by simple Read and Write exchange operations of small data files. There is no needed to access the source code of the program. The optimization is performed by a DAKOTA input file, in which all the necessary information is entered. The pre-and post-processing is a separate process, which starts and finishes with DAKOTA. The creation of the optimization loop of the part, pre-processing to post-processing has to be done by the user. The simulation can be very flexibly designed by the user. DAKOTA is a parameter file, and ends the simulation results in the results file. This loop continues, until an ending condition of optimization software is reached.

### 3.5.2 The input file format

DAKOTA will have an input file, with optimization settings. This file is read when the user starts DAKOTA from the command line as a parameter. The input file must be in a defined format. The file is divided into several blocks; each block contains certain settings for the optimization's implementation, see Table 1.

Table 1 Structure of a DAKOTA input file

| **block** | **parameters** |
|---|---|
| strategy, | |
| | single<br>graphics<br>tabular_graphics_data |
| method, | |
| | moga<br>output silent<br>seed = 10983<br>max_function_evaluations = 2500<br>initialization_type unique_random<br>crossover_type shuffle_random<br>num_offspring = 2 num_parents = 2<br>crossover_rate = 0.8<br>mutation_type replace_uniform<br>mutation_rate = 0.1<br>fitness_type domination_count<br>replacement_type below_limit = 6<br>shrinkage_percentage = 0.9<br>convergence_type metric_tracker<br>percent_change = 0.05 num_generations = 40 |
| variables, | |
| | continuous_design = 24<br># cdv_initial_point 0 0 0<br>cdv_lower_bounds 52 60 35 43 58 33 31 57 42 33 18 19 33 63 45 39 37 29 62 60 45 60 57 32<br>cdv_upper_bounds 72 80 55 63 78 53 51 76 62 53 38 39 53 83 65 59 67 49 82 80 65 80 77 52<br>cdv_descriptors = 'Y1' 'Y2' 'Y3' 'Y4' 'Y5' 'Y6' 'Y21' 'Y22' 'Y23' 'Y24' 'Y25' 'Y26' 'Y81' 'Y82' 'Y83' 'Y84' 'Y85' 'Y86' 'Y91' 'Y92' 'Y93' 'Y94' 'Y95' 'Y96' |
| interface, | |
| | fork<br>analysis_drivers = './module2-3-4_run.bat'<br>parameters_file = 'params.in'<br>results_file = 'results.out'<br>aprepro |
| responses, | |
| | num_objective_functions = 2<br>response_descriptors = 'obj_fn_1' 'obj_fn_2'<br>no_gradients<br>no_hessians |

### 3.5.3 Strategy description

The strategy section in a DAKOTA input file specifies the top level technique which will govern the management of iterators and models in the solution of the problem of interest. In this block, the optimization strategy can be defined. It is possible to combine a global search algorithm, through the use of multiple algorithms, one for global optima with an algorithm for subsequent local search. The algorithms must be nested. Mutli-level uncoupled method list a number of optimization algorithms in use, required in the entered order of processing. If a single algorithm is used, the names of the algorithms are no necessary and the entry can be omitted.

### 3.5.4 Method description

The method section in a DAKOTA input file specifies the name and controls of an iterator. The terms "method" and "iterator" can be used interchangeably, although method often refers to an input specification whereas iterator usually refers to an object within the iterator hierarchy. A method specification, then, is used to select an iterator from the iterator hierarchy, which includes optimization, uncertainty quantification, least squares, design of experiments, and parameter study iterators. In other words, the method is a model with no further need for a definition of an optimization strategy. The block is for such a model is defined by the expression “Single”.

### 3.5.5 Model description

The model specification in a DAKOTA input file specifies the components to be used in constructing a particular model instance. The model provides the logical unit for determining how a set of variables is mapped into a set of responses in support of an iterative method

### 3.5.6 Description of variables

The variables section in a DAKOTA input file specifies the parameter set to be iterated by a particular method. This parameter set is made up of design, uncertain, and state variables. Design variables can be continuous or discrete and consist of those variables which an optimizer adjusts in order to locate an optimal design. Each of the design parameters can have an initial point, a lower bound, an upper bound, and a descriptive tag.

The entries for a continuous design are the type and number of variables used. The name continuous defines the variables as real numbers, and the definition of integers relates to discrete design. The entries cdv-initial point, cdv-lower_bounds, cdv-upper_bounds give the start value and the lower and upper limit of the variable value. The variable must be complemented by the entry cdv-descriptor, the name of the variable.

### 3.5.7 Interface description

The interface section in a DAKOTA input file specifies how function evaluations will be performed to map a set of parameters into a set of responses. Function evaluations are performed using either algebraic mappings, interfaces to simulation codes, or a combination of the two. The interface to the simulation is set in the block “interface”. The interface will be used to define input and output files as well as the analytical program used for the implementation of the pre-and post-processing and the implementation of the simulation as response. The file Analysis-drivers is the pre-and post-processing and initiation of the implemented simulation. The parameters-file is produced by DAKOTA and contains the data required for pre-processing. The simulation result is sent to DAKOTA after the post-processing file of the results.

### 3.5.8 Description of responses

The responses specification in a DAKOTA input file specifies the data set that can be recovered from the interface after the completion of a "function evaluation." Here, the term function evaluation is used somewhat loosely to denote a data request from an iterator that is mapped through an interface in a single pass.

## 3.6 Chapter conclusions

In this chapter, a bibliographic introduction to the optimization methods and more particularly to the genetic algorithms is presented. An explanation of the reasons on why it is necessary to use a stochastic technique of optimization is given. The history and origins of evolutionary approaches is described, and so is an introduction of Darwin's theory of natural selection. The derived concepts from biological evolution that are being implemented in computers are explained: a) the unit of variation and how the genes behave to pass information to their successors; b) the unit of selection, that explains the fitness mechanism; and c) the unit of evolution that relates to change over time and iterations. A more detailed explanation of genetic algorithms is given when the chapter describes its use in simulation and the optimization loop is graphically presented. Also, definitions on the basic terms used in the genetic algorithms are shown. In order to exemplify how the genetic algorithms in simulation perform, a simple mechanical problem is presented related to the topology optimization of a structural bridge.

It is also introduced in the chapter a bibliographic analysis of multi-objective optimization with the use of genetic algorithms. It is explained how most of real problems are inherently multi-objective and how previous approaches have been used in the solution of such cases. The concept of Pareto frontier is explained and how the Pareto graph can be used to reveal conflicts in performance parameters of a system. A brief mathematical description of multi-objective optimization problems is presented, together with some basic definitions and approaches. Finally, an explanation of the optimization software used during most of the deployment of the case study of this research is given.

## Chapter 4 Development Strategy of the Crankshaft

The following is a procedural framework for extending the optimization paradigm into an innovation strategy to be applied in a general crankshaft development method. The strategy has different perspectives and it could be used to a more general range of applications. The main scheme, which relates to the steps of a product development cycle [83], is enhanced and consists of the next activities.

*Definition of the innovation problem and identification of the requirements or functional parameters in conflict*. During this step a deep understanding of the problematic situation has to be done. It is worth noting that at this stage on the development of a product, a preliminary design concept should be finished. A model a the process of conceptual design can be found in [84]. Functional relationships between components of the system has to be completely understood by the design team involved in the project. During this stage of the development, the functional conflicts (contradictions) are identified and the corresponding parameter metrics can be measured.

*Construction of the computer aided design and engineering (CAD/CAE) model*. The use of computational tools for the representation of the geometry and functional behavior of a product or system nowadays more frequently used in earlier steps of the product development cycle. The construction of the model should include the geometric parameters and physical phenomena that the behavioral simulations are going to perform. In the case of the present research with the crankshaft, the geometric modifications, balancing calculations in the form of embedded equations and finite element analysis to simulate dynamic behavior were represented in the model. During this stage, it is required that not only a complete single run of the simulation be performed, but a sensitivity analysis of the parameters involved. As described in the last chapter, a sensitivity analysis for the crankshaft was performed in order to identify the geometric characteristics that have the greater influence in the functional characteristics that compose the fitness functions.

*Development of the optimization/innovation task*. This stage corresponds to the main content of the research work and includes the activities described in the following sections of the chapter. Is during the development of these activities where the combination of elements that are part of the research, even when alone are commonly used, produce the outputs than extend a normal optimization result into a more innovative paradigm, as described in the introduction of the research. The specific requirements to be satisfied by the strategy are:

- An in-depth study of optimization techniques. What are the parameters and objective functions implied to evaluate the behavior of the crankshaft? Response surface methods are suitable for problems with several variables and one or more response variables, like crankshafts, even when it is little known about the process. Besides, some part of the methodology approaches multiple responses

- Develop interface programming that allows for integration of the different software: CAD for modeling and geometric evaluations, CAE for simulation analysis and evaluation, and genetic algorithms for optimization and the search for alternatives.
- Approach the target of imbalance of a V6 engine crankshaft, without affecting either its weight or its manufacturability.

*Analysis of results and synthesis of solutions*. The final stage of the general approach consist of an analysis of optimization/innovation results and a definitive and exhaustive inference of the design team into the concepts obtained by the algorithms. No computer substitutes the imagination and creativity of a person so the final decision about the configuration of the product remains in the team of engineers. What remains in the analysis is to find the ideal final result (TRIZ nomenclature) of the strategy and express it as an additional set of requirements (explained in the last chapter of the research):

- Find the innovation principles that complement the genetic operators (mutation, crossover, etc) to be able to generate a list of "extended operators" to apply to the geometry.
- Obtain new design concepts for the shape of the counterweights that help the designer to develop a better crankshaft in terms of functionality more rapidly than with the use of a "manual" approach.

The previous steps can be identified in a standard product development cycle, where the elements of the research strategy are contributing the most. The new element can be viewed as an integration loop that merges the activities during concept generation and evaluation, together with detailed design activities, as shown in Figure 4-1.

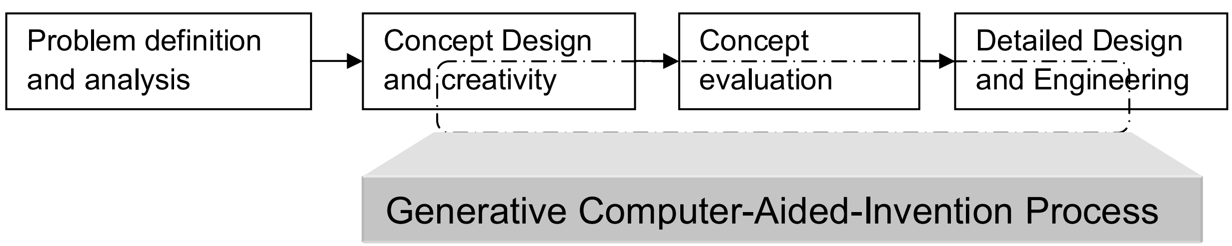



**Figure 4-1 Integration of the optimization/innovation algorithm into the development cycle**

The general procedure of the strategy is the integration loop into the framework, which compiles the elements that are part of the optimization tasks. For defining the problem, the crankshaft optimization task of this research work is converted into a parameter value optimization task by using spline curves for shape representation, particularly, the counterweight shape is represented by a spline curve. The model geometric variation of the shape is defined by floating-point values assigned to corresponding control points of the spline curve. Thus, the coordinates of the spline curve control points are serving here as design variables. To optimize the shape, the values assigned to control points must be optimized. Then, in principle at least, any non-linear parameter optimization method can be used to solve this optimization task and thus optimize the shape of the counterweight cross-section. However, in practice a global optimization algorithm like GA, appears to be essential for obtaining solutions of acceptable quality.

During the optimization loop, the coordinates of the control points that define the splined profile of the crankshaft modeled in the parametric CAD software is automatically manipulated by the Genetic Algorithms thanks to an interface programmed in JAVA language and controlled by an optimization algorithm, i.e. by a genetic algorithms (GA). The JAVA interface allows the CAD software to run continually with the crankshaft model loaded in the computer memory, so that every time an individual is generated the geometry automatically adapts to the set of parameters. The splined profiles allow shapes to be changed by genetic algorithms because the codified control points of the splines play the role of genes. The equations of imbalance are introduced into the CAD software, and automatically provide the value of the fitness function, i.e. the difference between the current and target imbalance. This value is automatically updated every time the geometry is modified, because of the interface linking the CAD model to the genetic algorithms that control the optimization process.

The tasks described next are intended to be automated, so the optimization loops could be programmed to run with a close monitoring of the results, produced in every evaluation. In the first place it is described the structure of the task for the optimization of imbalance. The second task describes the multi-objective optimization between the imbalance and the curvature. The third task describes the multi-objective optimization between imbalance and eigen-frequencies.

## 4.1 First Task: Geometric Optimization of Crankshaft's Balance

The first task involves the optimization of crankshaft's imbalance considering only geometric restrictions related to the profile of the counterweights. The use of genetic algorithms in the design of balancing of forged crankshafts is based on the premise that the counterweight´s profile shape can be modified (inside customer specs) to have a robust balance as an output function to be evaluated. This can be attained by following the general criteria described below:

a. *Identification of the control parameters*. Experiments will be performed on the geometry of the counterweights to find and confirm which counterweights have more significance to the balance of the crankshaft. Once done, the profile of the significant counterweights will be controlled by spline points.
b. *Simulation of balance*. A simulation program for the balance of the crankshaft will be run to have a measure of the influence of the counterweight´s profile changes. It is required to find an interface of the CAD software to the simulator. An evaluation criterion is required for any change of the significant counterweights.
c. *Codification of the parameters*. A chromosome representing the control points of the spline or control parameters to be evolved will be developed. In addition, an interface of the CAD software to the codification program needs to be developed.
d. *Runs of different optimization algorithms and comparison between them*. Why are genetic algorithms preferred? What advantages do they have in contrast with the classic

optimization solvers (e.g. steepest descend, direct search)? What is the advantage of using new GAs heuristics that deal with multiple objectives that contradict each other?

e. *A study describing differences between optimization tools.* The advances in the automation of the evaluation runs to improve a crankshaft's balance, from the equations embedded into a CAD model, are presented.

The proposed sequence of steps can be observed in Figure 4-2, which shows the parallel execution of DAKOTA and CAD software. The CAD software initialization loads the crankshaft CAD model into memory which, as explained before, needs to be pre-processed with all the codification and functional implementation. Parallel to the CAD initialization the DAKOTA software is initialized, which will activate the optimization algorithm, as explained in the previous chapter. The communication between the genetic algorithm and the CAD software is performed by an interface programmed in JAVA language that is triggered every time the coordinates for the modification of the splines for a particular individual appears. JAVA conducts the regeneration of the CAD model of the crankshaft and returns the imbalance results to DAKOTA. The detailed procedure will be described in the next chapter.

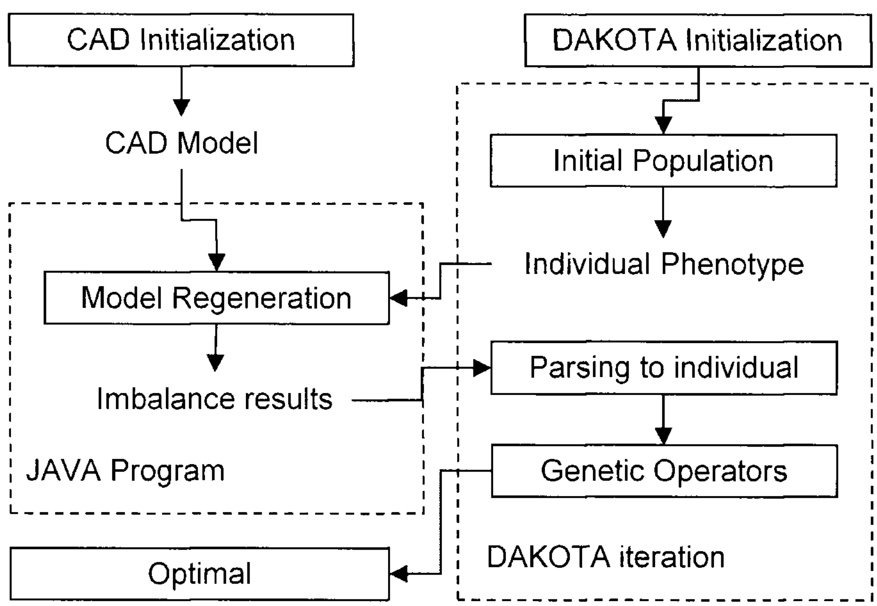



Figure 4-2 Integration of DAKOTA with CAD Software using Java

## 4.2 Second Task: Multi-objective Crankshaft Optimization

The activities for the second task define a research topic in multi-objective system optimization, focused on the development of engine crankshafts. From the results attained from the first task on optimization of crankshaft's balance, based mainly on CAD geometry modifications using genetic algorithms, activities for the second part are extensions of the general strategy. Objective functions and restrictions are added by the use of FEM simulation of the dynamic response during geometry variations, resulting in what attempts to be an integration of different systems running totally or partly automatically.

From the first task, it was found that additional objective functions and restrictions needed to be considered as important in the development of the research strategy. The restrictions are defined as follows:

a) Balance of the finished crankshaft (Restriction)
b) Dynamic response of the geometry: Eigen-frequencies (Objective or restriction)
c) Minimization of the mass of the crankshaft (Objective)
d) Curvature of the geometry of counterweights (Restriction).

Those four objectives-restrictions define a multi-criteria approach for the optimization of the system.

As mentioned in the previous section, a conclusive study on whether genetic algorithms are the best suited for this particular problem will be done. Meanwhile, the task scheme will be prepared, based on the assumption that GA´s are used (derived from the reference papers consulted) and, in case of a change to a better optimization algorithm, only the corresponding interface will be modified. Modularity of the scheme is encouraged in a such way that parameters and results are triggering, via text files, the systems execution.

For a consequent shortening of the product development process, the application of optimization processes and parametric geometry information are required within corresponding CAE processes. It would be obvious to use parametric finite element models (FE models) for the model generation in FE pre-processors. However, parametric modeling in common FE pre-processors is only supported rudimentarily. Therefore, parametric modeling within FE pre-processors can be very time consuming and is only feasible for a certain kinds of parameters.

Since the use of parametric design of CAD models has been established in common CAD environments, it seems natural to also use these existing parameters for CAE processes. This can be achieved by the integration of CAE functionalities in CAD environments. Especially CAD tools have been enhanced by analysis functionalities in the last few years and it is now possible to conduct parameter studies directly in CAD environments. However, at the present time, this is mostly restricted for use in internal FE solvers. Furthermore, the functionalities supplied in parameter studies and optimizations cannot compete with the possibilities of dedicated software packages like DAKOTA.

To use the parametric geometry information from CAD models in combination with external CAE software, it is necessary to set up a continuous automated process, starting from the CAD model and ending with the result evaluation of the analysis. Modern CAD and optimization software offer the basis for such an automated process, but they must be linked each other via suitable interface software.

The parametric generation of components in 3D CAD software has been established in many industry sectors, but no company in particular has yet achieved a continuous use of these models. Some CAD software offers possibilities to generate 3D volume meshes based on CAD models and to define boundary conditions and loads for a FE analysis. Commercial software also suppliy integrated solvers to conduct FE analyses. However, the utilization of the widely spread solver MSC.Nastran was preferred for the processes described in this section. For some CAD software there is the option to export MSC.Nastran models directly from the CAD environment. These models do not need any further processing before calculation.

This section extends the work done in the previous one, where the fitness evaluation consisted of the crankshaft imbalance. An evaluation of the first Eigen-frequency of the crankshaft was added. This evaluation is performed for every geometrical configuration of the CAD model by external CAE software. This two-objective optimization problem has a non-trivial Geometrical solution, owing to the different nature of the objective functions, one being in the frequency domain (eigen-frequencies) and the other, in the geometric domain (imbalance). The crankshaft structural analysis to be used as an additional objective function, predicts the crankshaft dynamic response based on the finite-element method. The additional fitness-function for the GA is based on evaluation of the simulation results because the engine crankshaft is a finely optimized component with significant resonances (both torsional and bending) within its normal operating range. The additional objective is the first eigen-frequency, which has to be increased. With this approach, objective and constraint functions involved can be evaluated either on the basis of the FEM simulation results or directly on the basis of the geometric shape itself, when the counterweight's shape is being modified.

The dynamic optimization of a crankshaft can be understood as an optimization problem with two distinct and, perhaps, conflicting objectives. It also involves the unpleasant features of crankshaft eigen-values that cannot be explicitly written but can only be calculated by simulation of the system model of the CAD geometry. In addition, constraints in the design's parameters should be taken into consideration.

The first step is defining the representation of the component within CAD software. Besides common CAD-geometry, boundary conditions, loads and properties of the model have to be defined. The actual mesh has to be defined by element sizes, edge divisions, fixed points, and curves.

The sequence of steps for this task is very similar in construction than in the previous task: The CAD software and DAKOTA execute in parallel. DAKOTA passes over the new set of parameters to the CAD software, which regenerates the model, exports the imbalance result and initiates the mesh generation. After the FEM mesh is created, Java starts the MSC.Nastran analysis and performs an evaluation of the results obtained by the analysis. After finishing the analysis, the Java application evaluates the analysis results and send the results, together with the imbalance results, to DAKOTA. The process is illustrated in Figure 4-3.

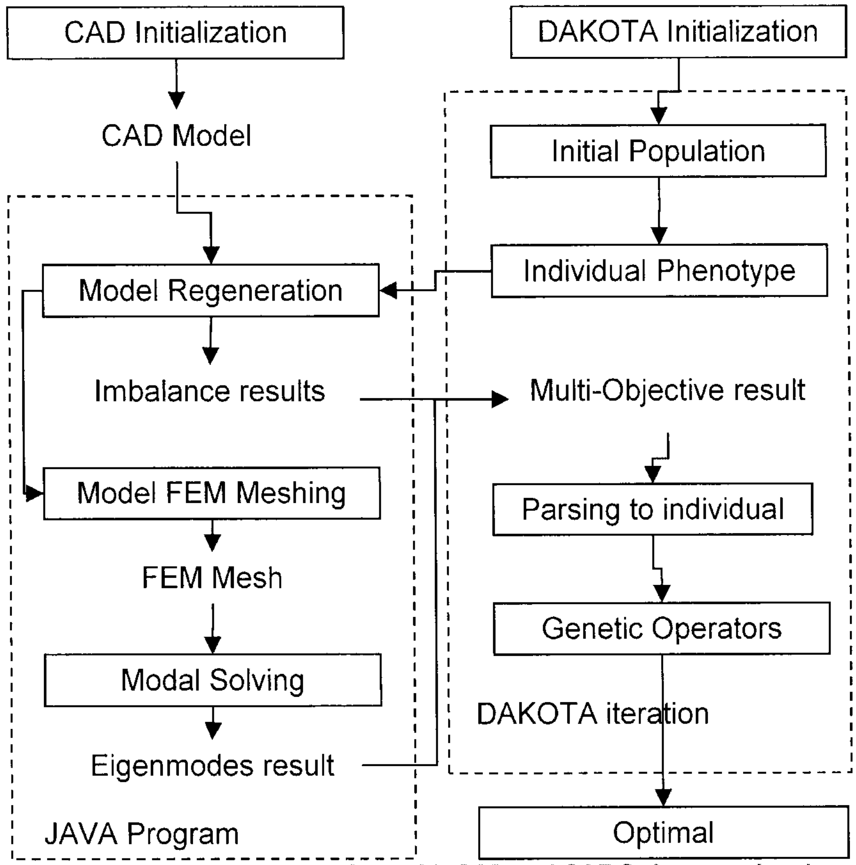



Figure 4-3 Integration of DAKOTA with CAD and CAE Software using Java

## 4.3 Third Task: Forging Optimization of Crankshaft's Dies

The application of Genetic Algorithms to the tooling design is an approach intended to be made on a similar strategy but different environment and platform. Non-linear FEM simulation software can be used to perform computational analysis of the forging transformation process. Evaluation of the results is made according to different output parameters, mainly: load, deformation flow, die strain and breakage, etc., obtained with the simulation.

a) *Input parameters are the shape of the forging tooling, made in CAD systems.* So development of control points over the dies to modify geometry requires interfaces with the programmed genetic algorithms.
b) *Simulation of forging.* It requires considerable computational capacity due to the plastic deformation of the forging process. One commercial software application has the capability of use parallel net processors to run faster. This is advantageous when genetic algorithms are conducting in parallel the evaluation of individuals. But the research in this topic deserves a separate chapter.
c) *Topological variation.* Codified as chromosomes, the control points of the working surface of the dies are going to be altered by topological variation, making the geometry of the tooling go in directions not considered by the designers.

At this point of the research, it was possible to reach the proof of concept for the new counterweights with splines. A simulation of the forging in comparison with the original profile was made.

Other approaches for performing simulation have been made in [85] which a simulation tool known as the upper bound elemental technique was used. That technique embodies the characteristic of fast simulation and can be used in 3D and for preform design.

## 4.4 Chapter conclusions

This chapter describes the general strategy and framework from which the optimization paradigm is extended into an innovation approach. The general framework enhances a standard product development cycle by included an optimization/innovation integrating loop. The steps of the strategy are described, beginning with the definition of the problem, the construction of the model, the development of the optimization/innovation task and finally the analysis of results and synthesis of solutions. An explanation of how the activities of concept design and evaluation can be merged with the activities of engineering design can be integrated is given.

It is explained how the different tasks will connect the genetic algorithms to the CAD model of the crankshaft, and how the splines will be modified according to the corresponding individuals thanks to an interface programmed in Java language. The first task relates to the optimization of crankshaft balance. The characteristics and conditions for the loop are explained. The second task relates to a multi-objective optimization involving not only imbalance but dynamic analysis with FEM. The integration is made in a similar way than the first task but is extended to include CAE software for FEM analysis, which makes the task more compiling and implies the conditions for a contradiction to appear. The detailed description of the tasks are given in the next chapter. Finally, the conditions required for a third task on optimization of crankshaft's dies are presented but the extent of the research only covers the proof of concept.

# Chapter 5 Deployment of the Implementation

In this chapter a description of the construction and set up of the model in the CAD software is presented. The CAD software is a commercial alternative called PRO/Engineer that is selected due to the programming functionality to be manipulated automatically. A description of the CAD model structure is followed by the transformation of the counterweights of the crankshaft from their commonly arc-shaped design. The curved profiles are substituted by splines, one of the main elements of the strategy. The imbalance analysis of the CAD model of the crankshaft and the formulas introduced in section 2.2.2 are embedded in the model, as long as the curvature and the eigen-frequencies. The control interface programmed in Java language and the requirements of the programming in order to let PRO/Engineer and nastran to communicate consist of a series of modules. The modules are a group of Java programs that allow to perform a series of commands upon the model of the crankshaft. The first module starts the execution of the PRO/Engineer in order to load the CAD model into the computer memory. The second module executes the regeneration of the model according to the data provided by the genetic algorithm. The third module is used during the multiobjective optimization between imbalance and eigen-frequency analysis. It produces a finite element mesh form the updated model and sends the mesh to the FEM solver.

The results of the tasks described in the previous chapter are an exemplification of how the general strategy can be applied. According to the strategy for the optimization/innovation of an engine crankshaft, the first two tasks, related to the imbalance and the eigen-modes, are integrating a series of activities that link the conceptual steps with the analysis steps of a typical product development process. Such integration enriches the production of designs.

## 5.1 Construction of the Model in CAD System

The CAD model is built using an assembly of several components. In 3D CAD systems for modeling parametric solids, the information is structured hierarchically. Shape and topological changes of the parts are performed to encourage changes in the existing relations between the features that constitute them. The geometric forms of the faces or surfaces, edges and vertexes are dependant on the parent features. Features are formed from sketches (the new geometry incorporated into the model) that are added by Boolean operations among its constituting features, or can be originated from the modification of created geometry (faces or edges) as it is the case of fillets, chamfers, shells, etc. The way to promote automatic changes in the shape of the parts is to introduce direct changes in the sketches of it. Those changes automatically originate topological and dimensional changes in the faces, edges and vertex.

The finished modeled crankshaft with consists of counterweights numbered W1 to W2, as explained in . For a better understanding of the modeling, the counterweighs that are going to be transformed with splined shapes are CW1, CW2, CW8 and CW9, that correspond to W1, W2, W8 and W9, as explained in section 2.1. In the pins numbered P1 to P6 are assembled the six additional elements known as fictitious weights. The fictitious weights are used to simulate the weight of the pistons. They are specified in the assembly drawing of the crankshaft. The balance analysis is a set of analysis features used to calculate the mass properties of the assembly. The components are the counterweights and will be placed under the assembly of the crankshaft. Figure 5-1 shows the complete model of the crankshaft assembly.

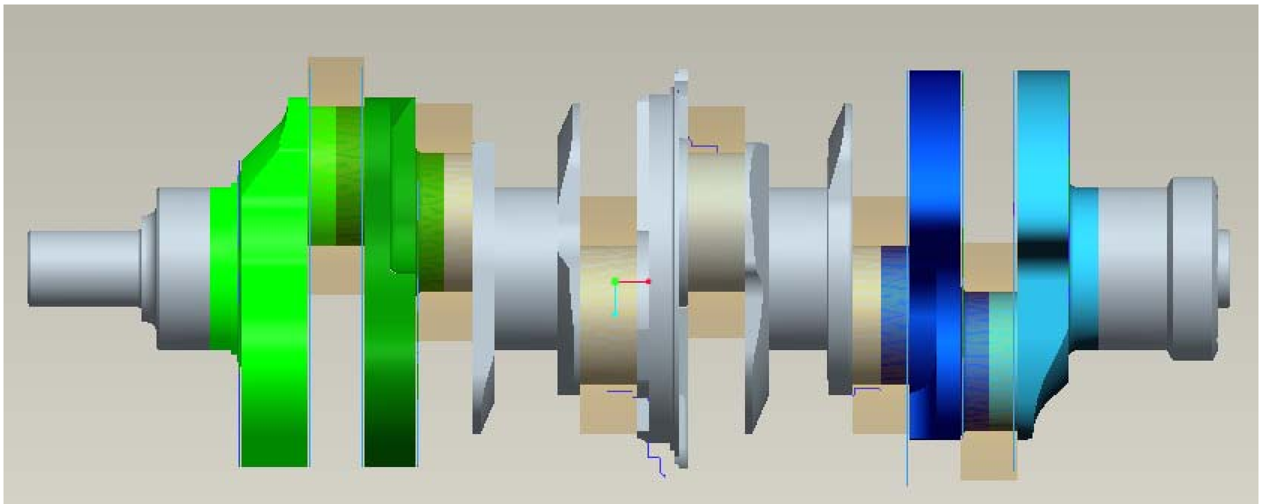


**Figure 5-1 Assembly model of the crankshaft**

The assessment of the design generated by the optimizer draft is done using a CAD system. The CAD system used in this method is Pro/ENGINEER version Wildfire 2. This software has the advantage of a JAVA based interface with which it can be almost completely controlled. The missing commands can be macros that can be accessed on the JAVA interface. However, in principle, it is possible the use of any CAD system, if the following conditions are met: Automation of the required functions, fault tolerance with regard to the input values and stability in the preparation of geometry. The assessment of the intended design can be reached by the geometry of the model. For this purpose, the weight of the crankshaft, the curvature of the counterweights and the balance is calculated.

After modeling, all the components are put into an assembly, and the required analysis features to calculate the imbalance are introduced. The last component is the machining model but for the purposes of the analysis, it is suppressed. The exact structure of the individual components and the assembly are described below.

### 5.1.1 Construction of the counterweights

The counterweights are modeled by three basic parts: the first contains the basic datum planes and the protrusions of the mains and pins plus the fixed base of the counterweights. The second consist of the variable portion of the CWs, which is formed by a spline with two fixed points (Point - A and Point E) at the beginning and end of the profile and three points along the length that are parameterized.

To execute geometry modifications the decision was made to substitute the current shape design of the crankshaft under analysis, from the original "arc-shaped" design representation of the counterweight's profile, to a profile using spline curves [28]. From the analyzed patents it can be noticed that any counterweight is constructed using a chained series of arcs. In Figure 5-2 our case study is shown how it is originally sketched. The substitution of arcs was done because, as already explained, cubic splines allow smooth shape changes via control points with continuous second derivatives, a desired property for material fluency during the manufacturing process.

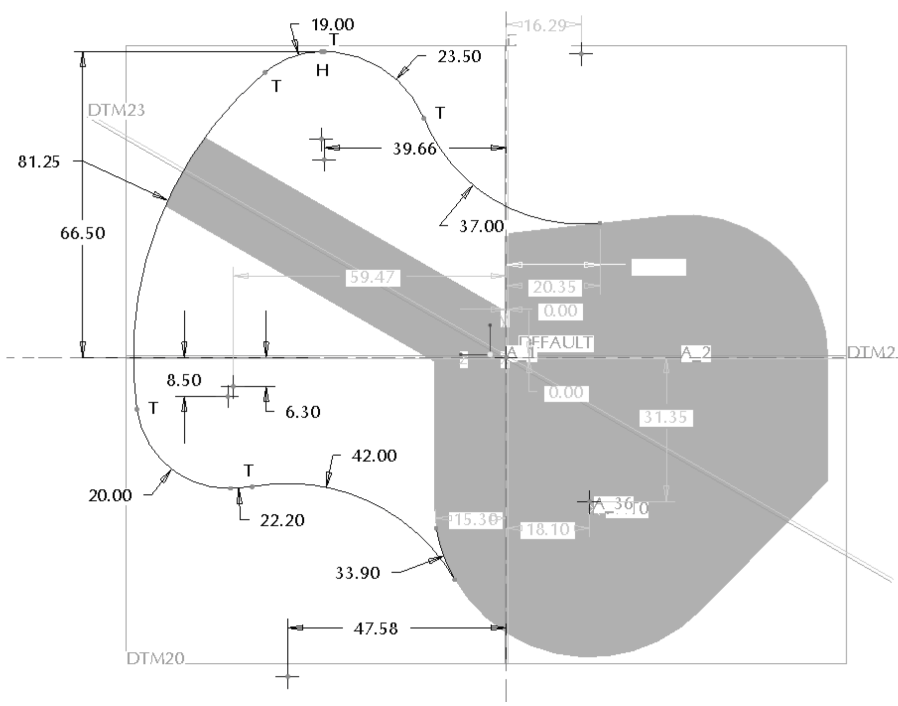



Figure 5-2 Profile of original counterweight number 2

From a previous sensitivity analysis it was found that defects in the profile (under-filling) of the outermost counterweights (CW1, CW2, CW8 and CW9), have the greatest influence on the balance of the crankshaft. Then it was decided to begin with codification of the "Y" coordinates of three control points, each upper and lower profile from these counterweights, adding 24 control parameters. The co-ordinates of the spline's control points are serving here as crankshaft design variables. Design variables are the quantities that are allowed to vary during optimization. The variation of the control parameters results in a balance response as design objective. Design objectives quantify the designer's desired properties of the array and require that metrics be defined that allow the genetic algorithm to evaluate the fitness of a particular design. To optimize the imbalance, the values assigned to control points must be optimized. Then, in principle at least, any non-linear parameter optimization method can be used to solve this optimization task and thus to optimize the design of the crankshaft. However, in practice a global optimization algorithm, like GA, appears to be essential for obtaining solutions of acceptable quality. The Figure 5-3 shows the counterweight's profile of the crankshaft with splines.

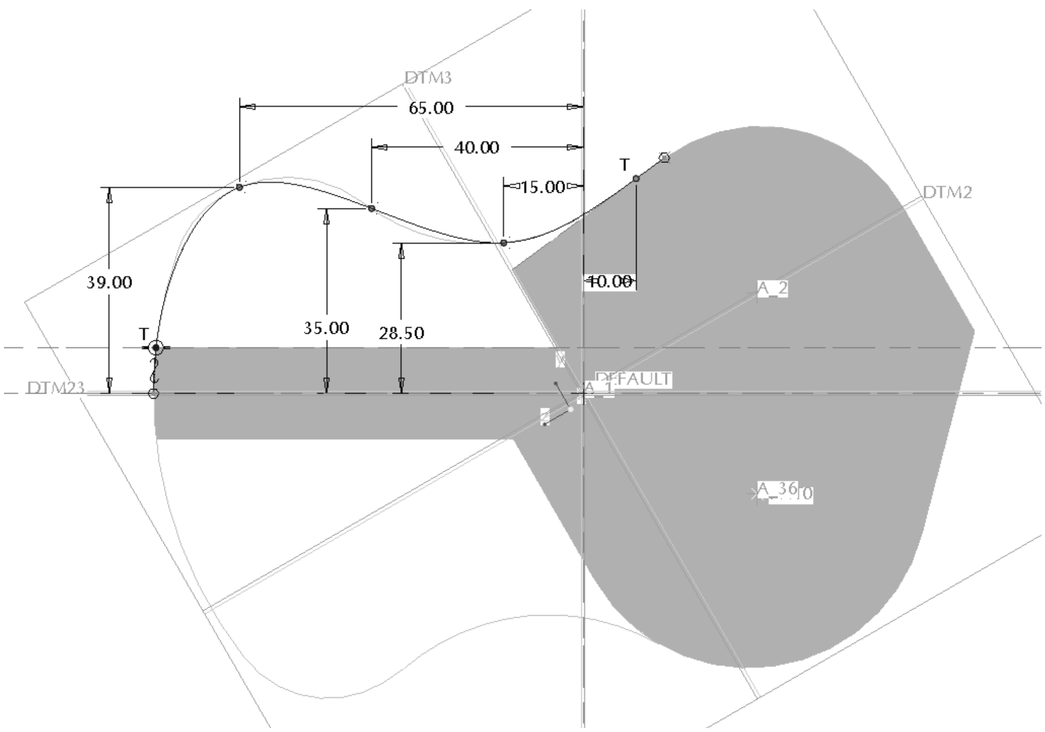



Figure 5-3 Profile of counterweight number 2 represented by a spline

The use of rectangular coordinates was decided of behalf of the general shape of the counterweight, but polar coordinates are also possible and they have been used in other works [86] . The last part includes some rounding of edges and the machining cuts. For the control of the profile shape, the coordinates of the points are parameterized by internal Pro/ENGINEER variables. Figure 5-4 shows internal variables driving the control points of the spline that define the profile of counterweight 1. These internal variables can be assigned on conditioned parameters, the parameters that the conditions on the coordinates of the points are assigned.

To make a curve shape without kinks and gradations of the counterweights, at the beginning and end of the spline the following is specified as a condition: a tangent to the continuity of surface of the partition. The profile was made using splines; and it would also train in the use of a design element. The counterweight was a full cross-section modeled with a thickness of 45 mm. The thickness of the counterweight can be altered eventually, but for the first trials was fixed, due to manufacturing restrictions.

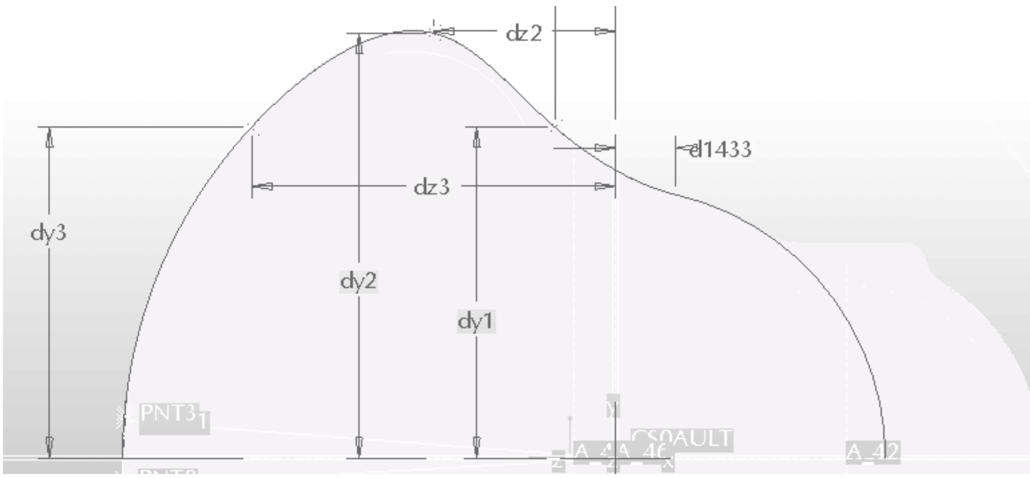



**Figure 5-4 Spline scheme of a counterweight CW1**

Figure 5-5 shows an interface window with the parameters shown in the profile and the relations introduced to the Pro/ENGINEER model of CW1, to control the spline profile that was shown in Figure 5-4. Dimensions named dy1 to dy6 are the y coordinates of the splines on both sides of the CW1. These dimensions are related to the Parameters *Y1* to *Y6* (current internal parameters of Pro/ENGINEER), which are assigned a value by the Java interface during an automatic control loop.

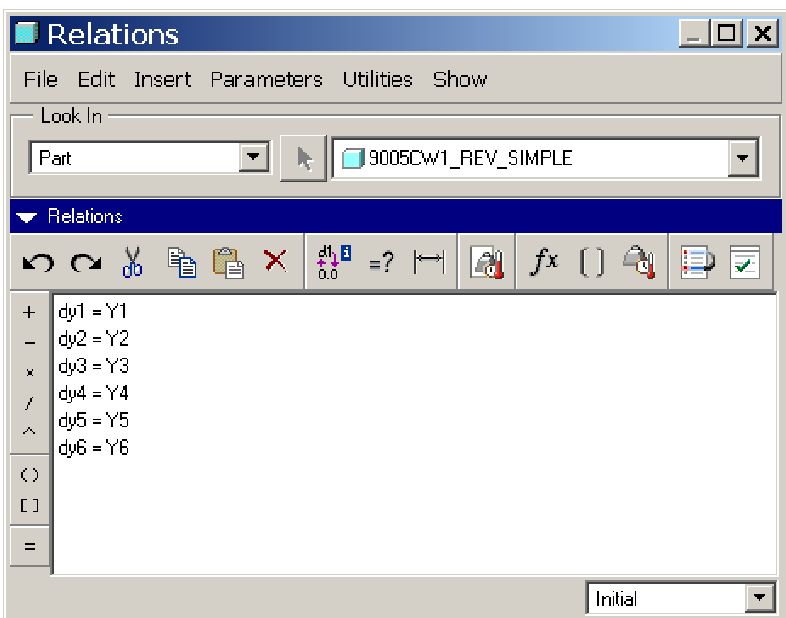



**Figure 5-5 Driving parameters of a counterweight**

The splining of the shapes and their control points, codified to be interpreted by genetic algorithms, form the basis for an evolutionary designed shape. Three main genetic operators act on the “genes” of the geometry: selection, crossover and mutation. Crossover allows the geometrical characteristics of selected splines (compared on the basis of a fitness function) to be merged in pairs and their properties to be extended to following generations.

The crossover has different levels of influence, the genotype and the phenotype (geometric coordinates of the spline’s control points), but its impact is in the level of the shape of the product. Moreover, the concept can be extended to the whole structure of the product and even to the functional structure. Figure 5-6 shows how the splines substitute the original profile of the other counterweights. The spline is within the tolerances of the original counterweight’s profile but will not be limited during the development of the shapes.

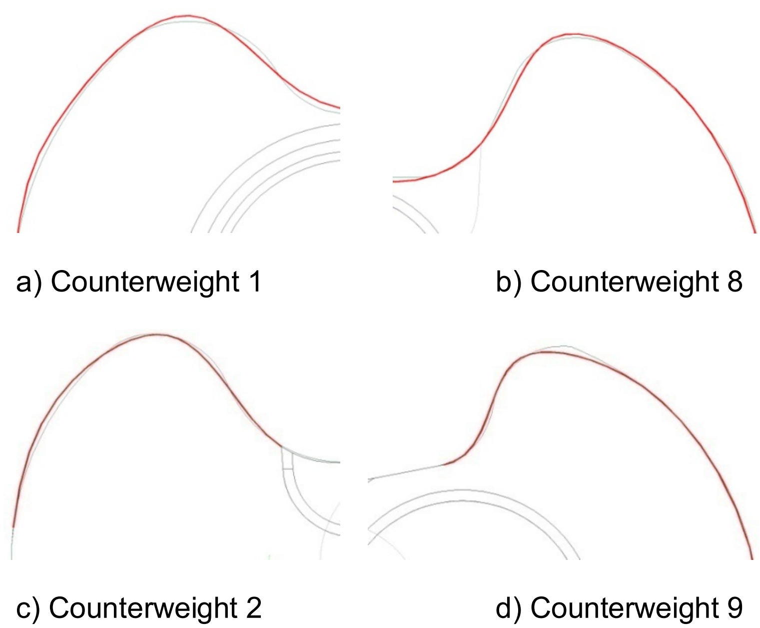



**Figure 5-6 Counterweights sketched using splines (dark lines).**

## 5.2 Imbalance analysis of the crankshaft

In Pro/ENGINEER the analysis features obtain the imbalance analysis of the crankshaft. The CAD model has to be in the same condition that it is when in the balancing machine, it is: completely machined, cold and with the fictitious weights properly located, and the system of units must be in mm and mass in grams. Then, these next steps must be followed:

1. *Create a coordinate system.* A reference system is located on the rotation axis and oriented with its x axis towards the flange end upon the rotation axis and the z axis in the proper vertical orientation (see Figure 5-7). The origin must be in the middle between the two correction planes.

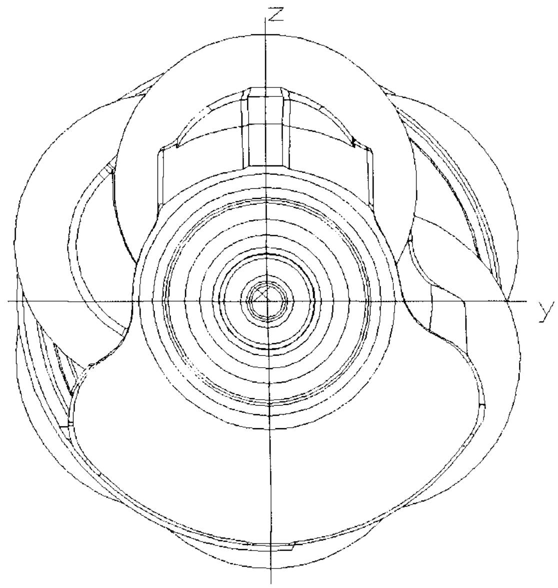



Figure 5-7 Orientation of coordinate system

2. *Obtain the mass properties.* PRO/Engineer, as most of CAD software, provides info with the proper command about the mass properties of the model. This data will be extracted from the model for the evaluation of the fitness function using the Java program. Using the coordinate system created in step 1 as reference, it is possible to obtain the required properties that are needed to calculate the imbalance. The crankshaft balancing is obtained by reading the mass properties from the model. As part of the model properties the following are included: Weight, Center of gravity and Moments of inertia. See Figure 5-8 for a data sample of the info file that is extracted from the model.

$m_g$ : mass of the machined crankshaft, including fictitious weights, identified in the figure as MASS.

$r_{gy}$ : y coordinate of the center of gravity of the crankshaft, with respect to coordinate system

$r_{gz}$ : z coordinate of the center of gravity of the crankshaft, with respect to coordinate system

$I_{xy}$ : the inertia product xy at center of gravity with respect to coordinate system

$I_{xz}$ : the inertia product xz at center of gravity with respect to coordinate system

The coordinates can be identified in the Figure 5-8 in the section CENTER OF GRAVITY. The inertia products can be identified in the section INERTIA at CENTER OF GRAVITY of the same figure.

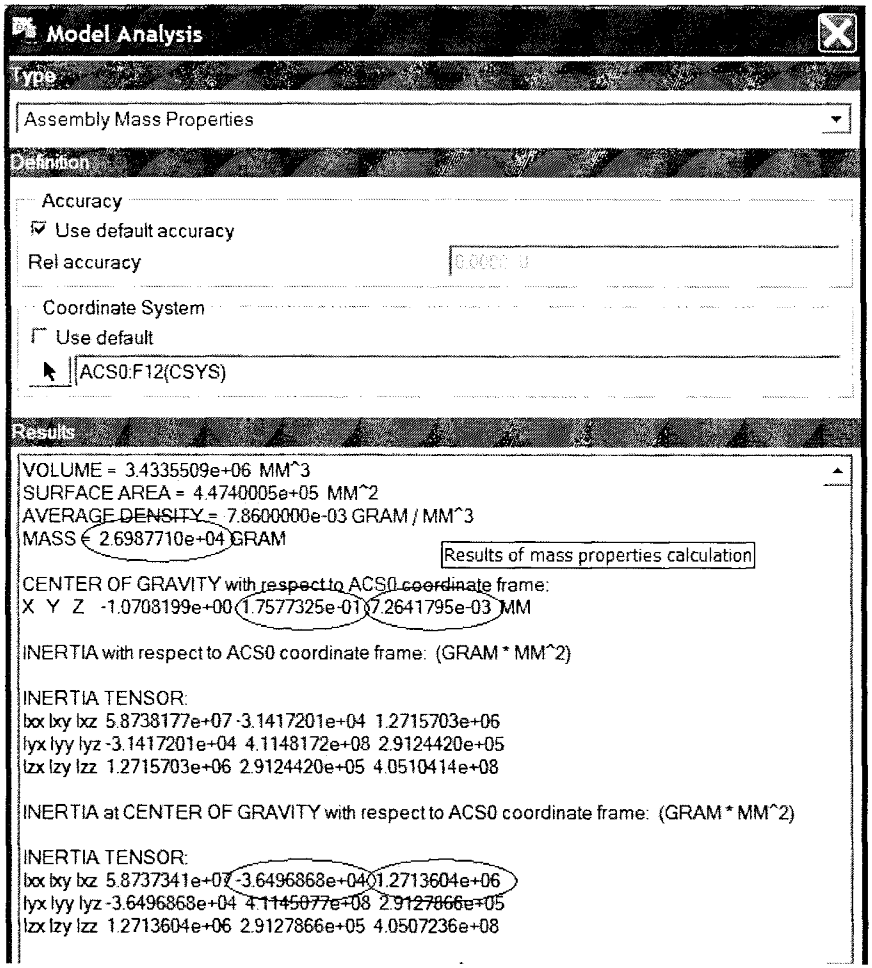



Figure 5-8 Mass properties extracted from CAD model

3. *Calculate the imbalance*. The current design unbalance is calculated according to the formulas deployed in section 2.2.2. The formulas are embedded in the CAD geometry of the Pro/ENGINEER software. With the embedded formulas, it is calculated the imbalance of the crankshaft in both correction planes. Therefore, every time the genetic algorithm produces a new individual, the model is modified in the counterweight profiles, and a new fitness value is returned to DAKOTA. Figure 5-9 shows the command window, called analysis feature, where the embedded imbalance formulas are introduced to the CAD software. It can be seen how the internal (codified) parameters related to the mass properties of the model, are assigned to a set of temporary parameters called *peso, IXY, IXY, YG, ZG*. Those parameters are used to build the formulas of imbalance $m_R r_{Ry}$, $m_R r_{Rz}$, $m_L r_{Ly}$ and $m_L r_{Lz}$ that, internally in the calculations, are composed by intermediate calculations that are identified as *MRRRY, MRRRZ, MLRLY* and *MLRLZ* in the Figure 5-9. Those values give the current imbalance of the individual ($m_L r_L$ y $m_R r_R$).

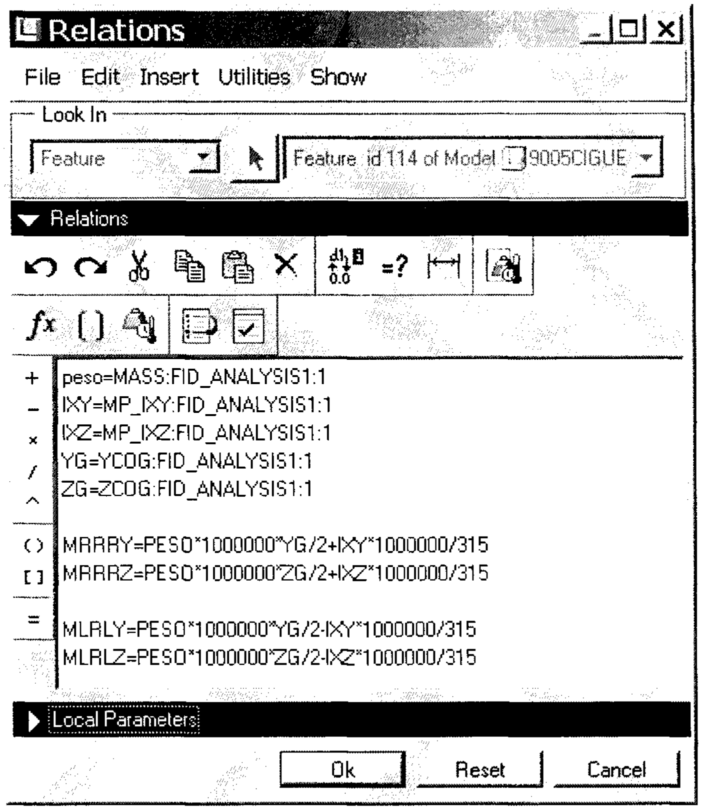



Figure 5-9 Embedded imbalance formulas in the CAD model

If a target for unbalance is specified, the difference between the target and the current unbalance is a set of vectors that will be minimized by making modifications in the crankshaft geometry. In the case of the crankshaft under study, the imbalance target is 400 g-cm in the *y* direction in both correction planes. Figure 5-10 (a) shows the analysis feature that calculates the function-to-target imbalance from both sides of the counterweight. And (b) shows the analysis feature that defines one of the fitness functions, relating the previous parameters of function to the target.

The function to target in the right correction plane is obtained using equations (1) and (2):

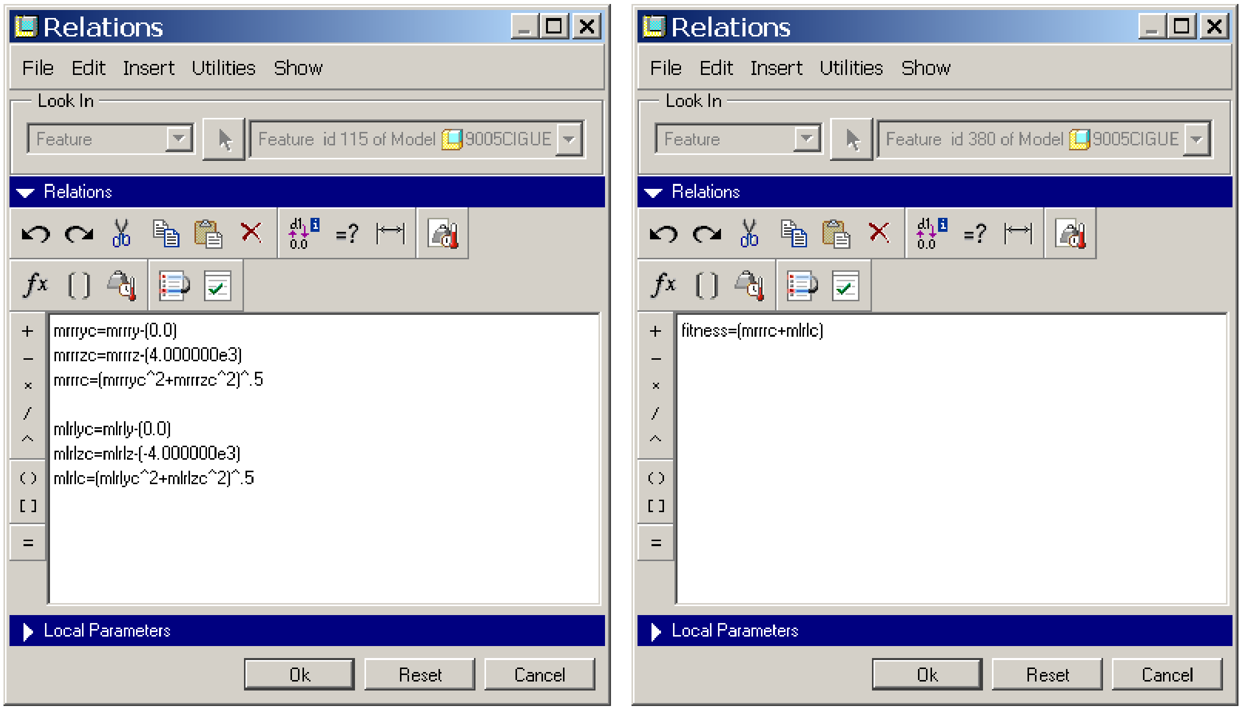
$$m_R r_{Rc} = \sqrt{((m_R r_{Ry} - 0)^2 + (m_R r_{Rz} - 4000)^2} \quad (6)$$

The function to target in the left correction plane is obtained using equations (3) and (4):

$$m_L r_{Lc} = \sqrt{((m_L r_{Ly} - 0)^2 + (m_L r_{Lz} - 4000)^2} \quad (7)$$

And the fitness function is the minimization of the sum of equations (6) and (7), according to the goal programming approach stated in equation (5), it is:

$$Fitness = \min(m_R r_{Rc} + m_L r_{Lc}) \quad (8)$$




(a) (b)

**Figure 5-10 Internal Calculation of the imbalance and fitness function in the CAD software**

During the calculation of the imbalance, problems often occur in Pro/ENGINEER such as large change in the profile that causes a possible collision and interferes with a different body in the same place. When this happens, Pro/ENGINEER cannot regenerate. Then a signal is provided, either a null or a NaN (Not a Number signal). This problem can also occur during a change of surfaces of different bodies. In this case, there is no other choice but to assess a fixed value or provide a signal null or a NaN to the fitness function. Optimization software cannot evaluate this design. This creates a great disadvantage for the evaluation of the design quality, as the individual can be greatly distorted by the specified imbalance. In the next section is described how the curvature of the counterweights of the crankshaft is obtained, directly from the CAD model.

### 5.2.1 Curvature of the counterweights

Curvature of the counterweights is a restriction that need to be considered because from it, it is inferred the forge ability of the crankshaft, as explained in section 4.2. The curvature of the counterweights is taken into account by calculating the maximum value in the profile. For an accurate assessment it is necessary, therefore, that the curvature of the entire profile be analyzed. For scanning the curvature, there are two options. One option can be using the curvature command included in the Analysis menu of Pro/ENGINEER. The problem with this option is that it cannot be embedded in an analysis feature.

The second option is to create a datum point on the spline and around this point create a circle curve from which a radius (and so a curvature) can be measured. Then a user-defined analysis feature is employed to scan the value of this circle curve along the spline and subsequently obtain the maximum value and extract it. Figure 5-11 shows the measure of the curvature used to illustrate the curvature along the spline, applied directly on the counterweight. It can be clearly observed that the Maximal curvature is at the bottom of the counterweight.

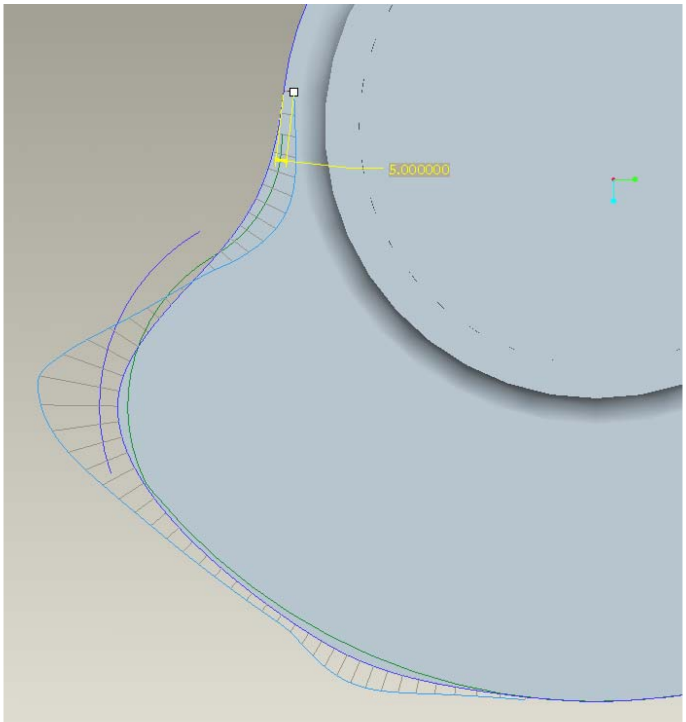



**Figure 5-11 Curvature of counterweight profile**

### 5.2.2 Analysis of the Eigen frequencies

It is relatively simple to define FE models from existing CAD models in Pro/ENGINEER. However, an appropriate parameterization from the constructional and manufacturing point of view is not necessarily suitable for an optimization or robust design analysis. It is usually necessary to adapt the parameterization and to consult with the design engineer in advance. This additional expenditure should already be considered in an early phase of the whole process. In some cases it is also necessary to generate additional geometry to determine the exact definition of the loads and boundary conditions. In the optimization processes, the FE mesh is regenerated during each calculation run. Thus, a control of the mesh is only possible via mesh parameters and not via manual intervention as in classical CAE processes. There are only restricted possibilities to control the mesh within Pro/ ENGINEER. Sometimes additional geometry must be used to generate comparable, good meshes in critical places. An example would be definition of additional curves as fixed curves in radii to guarantee comparable meshes.

To evaluate the eigen-frequencies of the crankshaft the procedure was to have the CAD model meshed in Pro/ENGINEER. For doing so, it is necessary to have a pre-modeled set of conditions available at PRO/Mechanica (from the Applications menu) and enabled to work with Pro/ENGINEER. If PRO/E fails to mesh a model, it is probably because PRO/Mechanica is not enabled. To do so, it is necessary to go to the installation directory of Pro/ENGINEER (C:\Programs\proeWildfire 2.0\bin) and run *ptcsetup.bat* . At the dialog box, any default option is reviewed until the windows: "*Optional Configuration Steps*" and Check: "*Configure Other Product Interoperability*" appear. The user clicks on Next and in the field "*Locate Other Installation Locations*" the installation location of PRO/Mechanica: (C:\Programs\mechWildfire 2.0) is introduced.

Three features need to be added: the material STEEL for every CW has to be assigned to every part individually. To every main rolling a cylindrical restriction is added, one of the four mains, additionally has an angular restriction and another main has an axial restriction. All four constrain completely the movement of the counterweight. Finally, a mesh control is assigned to all CW with a minimum mesh size of 2 mm.

## 5.3 Control Interface for the CAD Model

This section describes the control interface that is required in order to the Java program to manipulate the CAD software and send results to DAKOTA. Modern CAD software can be automated to perform with pre-programmed instructions. Pro/ENGINEER can be automated in three different ways. It offers advantages and disadvantages in every automation possibility and may have different features. To achieve a fully automated control of Pro/ENGINEER, a combination of two of the three options can be used.

### 5.3.1 Pro/PROGRAM

Pro/PROGRAM is a native programming language for the manipulation of complex relationships. The syntax of Pro/PROGRAM is similar to traditional programming languages, but with less features. The program is built directly into the CAD file and is embedded within Pro/ENGINEER. This means that user input to the implementation of optimization cannot be avoided. Due to this fact and the low functional scope, Pro/PROGRAM is not used in optimizing.

### 5.3.2 J-LINK

J-LINK is a powerful programming module for automatic control of Pro/ENGINEER. It is based on JAVA, and for this reason offers the same advantages as object-oriented programming and is platform independent. Thus, it is possible to build on an existing design, elements of access to it or create new ones. The functions of J-LINK are very large. However, some features, e.g. reading the curvature, are not implemented.

The interface offers a synchronous and an asynchronous operating mode. The source code is mostly interchangeable without problems between the two modes. A synchronized J-LINK program will be launched and executed directly into Pro/ENGINEER. An asynchronous J-LINK program works independently and is started and executed outside Pro/ENGINEER. The optimization was in the asynchronous mode.

For J-LINK to run, the Java Development Kit must be installed. This is done by download JDK from Sun Microsystems homepage and install to the programs directory. Then, a group of Windows Operating System environment variables need to be set up, these conditions are listed in Table 2

**Table 2 Environment variables required in the set up of J-link**

| Variable | Contents | Comments |
|---|---|---|
| CLASSPATH | .; C:\programs\Java\jdk1.5.0_10\jre\lib\rt.jar; | (1.5.0_10 can be replaced by newer versions) |
| | C:\ programs\ proeWildfire 2.0\text\java\pfcasync.jar; | (proeWildfire 2.0 can be replaced by newer versions) |
| PATH | C:\programs\Java\jdk1.5.0_10\bin; | |
| | C:\programs\ proeWildfire 2.0\i486_nt\lib; | |
| | C:\ programs\ proeWildfire 2.0\bin; | |
| | C:\ programs \proeWildfire 2.0\text\java; | |
| PRO_COMM_MSG_EXE | C:\ programs\ proeWildfire 2.0\i486_nt\obj\pro_comm_msg.exe; | |
| PRO_DIRECTORY | C:\ programs \proeWildfire 2.0 | |

### 5.3.3 Trail files

In Pro/ENGINEER software there is a macro-control called “trial file”. A trial file is a macro of a sequence of commands executed by the user to perform some activity with the software. It can be from any single command such as “copy and paste” to the modeling of a whole part. The sequence of commands can be recorded as an external file and later imported for use when the very same activity in the software needs to be made. It is very convenient for automation. In carrying out the trial file there are often difficulties because the references to objects have relative relations and are stored not by absolute names. It may occur that the model tree inadvertently changes during optimization and tries to execute on the wrong design element. The result is a stop during optimization, as the function is not applied on the desired design element. The errors can then often be resolved only by user input.

### 5.3.4 Combination of J-LINK and trail files

Since all the necessary functions for modeling a part are not available in the J-LINK interface, the combination with trial files is necessary. The combination at first glance looks easy, but the calls to the asynchronous mode of J-LINK interface cause some problems of coordination at the moment of executing commands. The interface sends the commands to Pro/ENGINEER without feedback on the state of execution. Particularly in the case of complex commands, sometimes the trail files executed over the J-LINK program are already responding to the next command before the current command is stopped. However, the J-LINK internal commands have to contend with insufficient time to be executed. To resolve this problem, at some points during the control, it is necessary to integrate a waiting period so that the execution is reliable. To perform reliable control some effort to operate at any point in the program is needed, because the waiting period required is large.

Pro/ENGINEER with Jlink offers a powerful interface for automation, which allows accessing most of the objects and methods of a session. This only applies, however, to the CAD part of the application. Within the integrated Pro/Mechanica environment, the automation can only be carried out with macros that repeat manually performed user actions, which nevertheless can be performed in batch-mode without a user-interface. This restricts the generation of a standardized control and results in an additional programming expenditure when a new model is being used.

### 5.3.5 Description of the J-LINK Modules

The J-LINK interface consists of four Java modules independent of each other but interrelated and coordinated by I-O text files (they communicate through files saved to the hard drive). The modules are divided into the following:

a) Pro/ENGINEER start up,
b) Update of the model,
c) Creation of the mesh and
d) Solving of the eigenvalues.

Each model is intended to work independently but interrelated. To have J-LINK running, it is convenient to add the file *pfcasync.jar* into the working local directory. The Java modules, as being programmed in Java, need to be compiled. To compile any of these four modules, it is necessary to run the corresponding *#_compile.bat* compilation file in the working directory. A manifest text file *#_manifest.txt* and A J-LINK library file *pfcasync.jar* have also to be present. The compilation file contains three lines:

a) *set classpath=".;C:\Programs\Java\jdk1.5.0_10\jre\lib\rt.jar;.\pfcasync.jar* This line resets the environment variable classpath in case it hast not been defined previously.
b) *javac module#_###.java* This line creates the executable .jar file from the corresponding .java file, and
c) *jar -cvfm0 module#_###.jar module#_###_manifest.txt module#_###.class pfcuParamValue.class* This line compresses the required files needed to execute J-LINK.

The manifest file contains the declaration of the Main class and the paths to the .jar files of the module and J-LINK. It consists of four rows:

a) *Manifest-Version: 1.2*
b) *Main-Class: module#_###*
c) *Class-Path: pfcasync.jar module#_###.jar*
d) *Created-By: 1.4 (Sun Microsystems Inc.)*

### 5.3.6 Pro/ENGINEER Startup

The first module is *Module1_startpro.java*. This module runs in a command window separately, after compiling according to previous instructions. The module consists of a Java program that starts Pro/ENGINEER and keeps it running while the optimization is executing. For a listing of the code, see appendix.

### 5.3.7 Update of the Pro/ENGINEER model

The second module is *Module2_proe_update.java*. This module runs when executed by the optimization loop, making connection with the module 1. It consists of three processes:

a) Retrieval of the parameters of the individual for an evaluation
b) Regeneration the model according to the parameters
c) Delivery of a geometric response, depending on the evaluation, which may be a balance response or a curvature response.

For a listing of the code, see appendix.

### 5.3.8 Meshing of the model

The third module is *Module3_proe_mesh.java*. This module runs in order to mesh the geometry of the model. The boundary conditions must be defined previously in the model. The meshing is performed after the regeneration, based on the individual in the previous model. For a listing of the code, see the appendix.

### 5.3.9 Solving for the eigenvalues

The last module is *Module2_nastranrun.java*. This module is run to perform the solution of the eigenvalue evaluation, as defined by the FEM mesh obtained in the previous module. This module takes the FEM file with the input conditions and executes nastran to create a solution file. The model also reads the required response and creates a results file for the algorithm to evaluate. For a listing of the code, see the appendix

**After passing the parameter sets to the CAD software and updating the geometry, the FE mesh has to be generated. This FE model generation with about 250,000 linear elements requires about 90 seconds at a workstation with 3 Ghz when Pro/ENGINEER is used. A summary of the files that must be in the working directory in order to the modules run is presented in**

Table 3

**Table 3 List of files required for the Java program to run**

| files | description | list |
|---|---|---|
| Java files: | Shown for one module. Other modules are similar (the .java files, and so the xxx _manifest.txt file, needed for compilation have to contain the | module1_startpro.class 6 KB<br>module1_startpro.jar 7 KB<br>module1_startpro.java 12 KB<br>module1_startpro_compile.bat 1 KB<br>module1_startpro_manifest.txt 1 KB<br>module1_startpro_run.bat 1 KB |

| | | |
|---|---|---|
| | path to the working directory) | |
| | auxiliary files: | -pfcuParamValue.class,<br>-pfcasync.jar |
| | needs to be edited to contain the path to the working directory | environment_modules.opt |
| Crankshaft files: | checks, main, pins and assembly, they must be done in a compatible Pro/ENGINEER | 9005cw1_rev_simple.prt.28 988 KB<br>9005cw2_rev_simple.prt.25 588 KB<br>9005cw3_rev_simple.prt.9 554 KB<br>9005cw4_rev_simple.prt.10 633 KB<br>9005cw5_rev_simple.prt.9 887 KB<br>9005cw6_rev_simple.prt.11 699 KB<br>9005cw7_rev_simple.prt.12 677 KB<br>9005cw9_rev_simple.prt.24 662 KB<br>config.win.7 6 KB<br>pin1.prt.13 55 KB<br>pin2.prt.11 55 KB<br>pin3.prt.9 56 KB<br>pin4.prt.10 59 KB<br>pin5.prt.9 55 KB<br>pin6.prt.11 55 KB<br>9005cw8_rev_simple.prt.30 615 KB<br>9005main_rev_simple.prt.35 869 KB<br>9005ciguenal_frio_simple.asm.90 238 KB |
| | auxiliary files: | -config.pro,<br>-config.win,<br>-schablonemodes.nas |
| DAKOTA Files: | | dakota_moga_v41_modules.in 2 KB<br>dprepro.pl 9 KB<br>geom_result.opt 1 KB<br>idr-parsed-input.txt 6 KB |
| | | param.opt 1 KB<br>param.opt.tpl 1 KB<br>params.in 2 KB |
| | trigger files: | -ProeRunning.txt,<br>-_solve_fertig.txt |

## 5.4 Results of the tasks

Next are presented the results of the tasks that include the optimization of imbalance and the multiobjective optimization including curvature and eigen-frequency. In each task the general conditions are provided in order to allow the repeatability of the experiments. However, in principle, no similar results are expected when repeating the run because of the random factor implied in the genetic algorithms.

### 5.4.1 Results of the optimization of imbalance

In this problem, a population of 50 individuals is generated randomly. Each individual is a gene containing a set of 24 parameters, codified in lineal mapping with 0s as the lower limits and 1s, the upper limits. Once generated, the phenotypes (de-codified) are written to a file and read by the CAD software to generate a fitness result. The parametric CAD software used in the project regenerates the model with an interface programmed in Java, and some of the subroutines were adapted to let this fitness result be read and sent back to the genetic algorithm, where it is assigned to the corresponding individual. The objective function was equation (8), as stated in section 5.2. After the whole generation has been evaluated, tournament selection plus other genetic operators produce the next generation. As an estimation of how long (elapsed time) a typical optimization exercise takes, the following elements are calculated: the automatic regeneration of an individual's geometry lasts about 15 seconds. When the regeneration of the geometry fails, it takes about 45 seconds for the CAD software to restart. The optimization run consists of about 1000 evaluations, of which approximately 10% of them fail the regeneration and return a NaN, so it is a 5 hours long run.

The genetic algorithm that was used in the very first runs was the vgGA. It performed well during the initial runs, as it was capable of delivering a geometry that improved the fitness function in each generation. Capable of running a substantial number of evaluations, vgGA showed a convergence to a fixed fitness value, as shown in Figure 5-12.

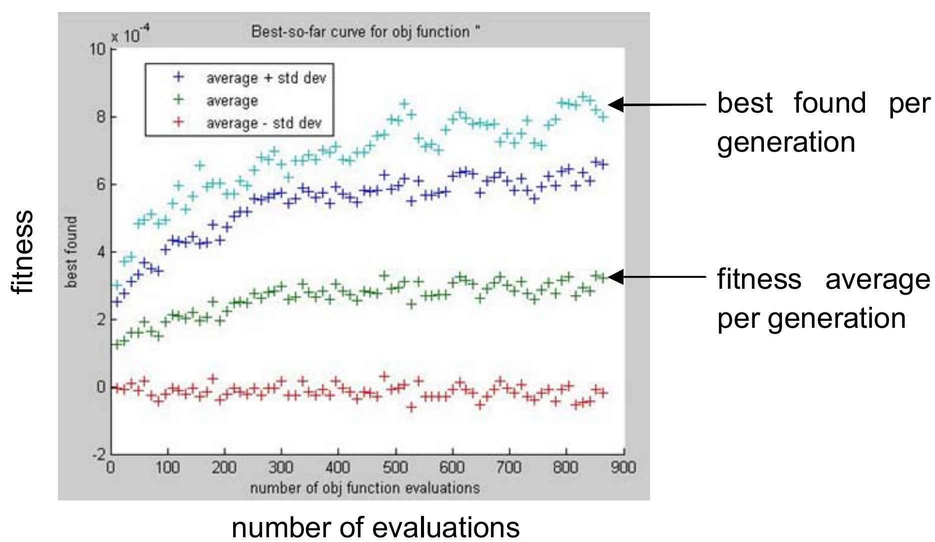



**Figure 5-12 Evolution of the fitness function using VGGA**

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. In Figure 5-13 It can be seen that the shape of some counterweights takes some non conventional forms like having notches.

It should be noted that the shapes, although not satisfactory from the forging point of view, tend to “separate” the balancing mass out from the crankshaft center axis.

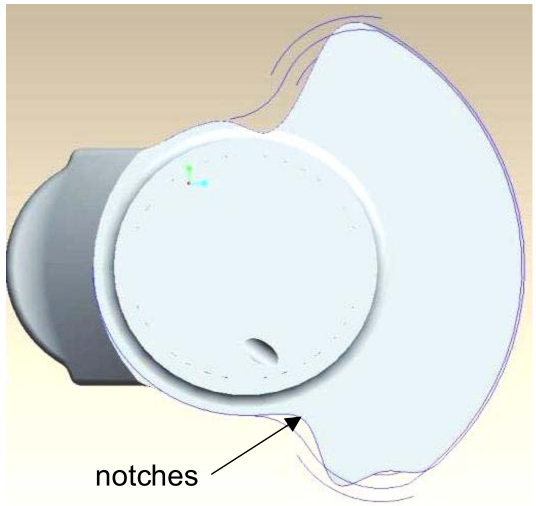



Figure 5-13 First shapes resulting from vgGA optimization

This behavior led 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.

In order to substitute vgGA with a DAKOTA GA, some adjustments to the strategy had to be made:

1. A DAKOTA GA, that solves for minimization instead of making a maximization, was chosen, 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) was from the DAKOTA “Stochastic Global Optimization” library, a genetic algorithm for real numbers.

In the first runs, it was not possible to reach the balance target inside the design constrains of the parameters that control the splined counterweight profiles. In Figure 5-14 the development of a fitness function can be observed. The x axis shows the number of evaluations; the y axis is the average fitness value of every generation.

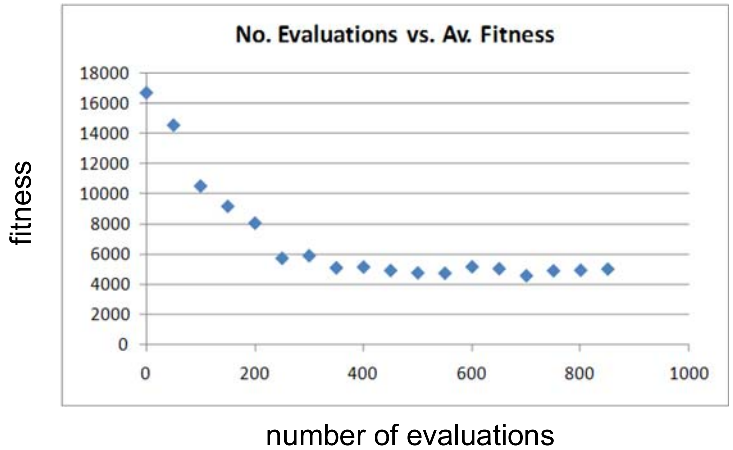



Figure 5-14 Generation history of imbalance fitness

No conclusive results are obtained when using a mono-objective function (fitness). After the balancing behavior of some individuals was analyzed, 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 a multi-objective approach.

### 5.4.2 Results of multi-objective optimization using imbalance in both correction planes as independent fitness functions.

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, further runs were performed. A multi-objective genetic algorithm (MOGA) was chosen from the DAKOTA toolkit to generate the required data. Some graphs from the analysis were obtained.

Figure 5-15 shows the distribution of the first generation in dots and the last generation in squares, with the value of the first function (imbalance of CW1) on the x-axis and the value of the second function (imbalance of CW9) on the y-axis. The results from the first attempts show that the imbalances from both sides of the crankshaft are in conflict with each other, according to the perceived distribution in the diagram. Nevertheless, the last generation show a movement towards the value of zero, which in this case is regarded as the "ideal point".

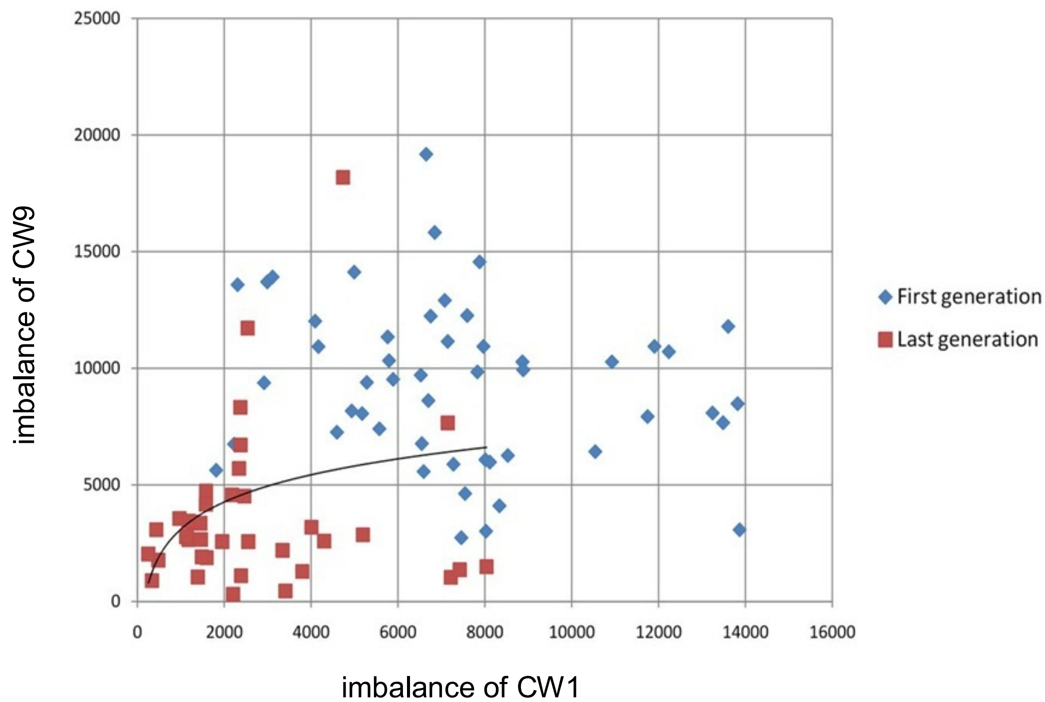



Figure 5-15 Graph of imbalance on CW1 vs CW9

It should be noted that the graph does not show a defined frontier, which would have been expected as the balance reaches the target. As the target balance could not be attained inside the design constraints defined for the geometry of the counterweights, some trials were performed to allow the spline to override the geometry constraints. However, although improved results were obtained, the convergence was not yet satisfactory. Results from previous attempts [49] where only two instead of four out of the nine counterweights had been splined, had not shown this behaviour.

From this last generation an individual was found to be very close to the imbalance target. In order to have a visual feeling about the way the algorithm was performing, this individual was modeled in the CAD geometry. The resulting geometry of that individual is shown in Figure 5-16. In these pictures the profiles of the counterweights are close to the geometric restrictions imposed by the original design. It is inferred that to reach the balance target, it may be necessary to reconsider these constraints. As can be observed in this case, sharp edges arise in the profile, which is not a convenient condition from the forging point of view of the crankshaft because this causes difficulties in the material flow.

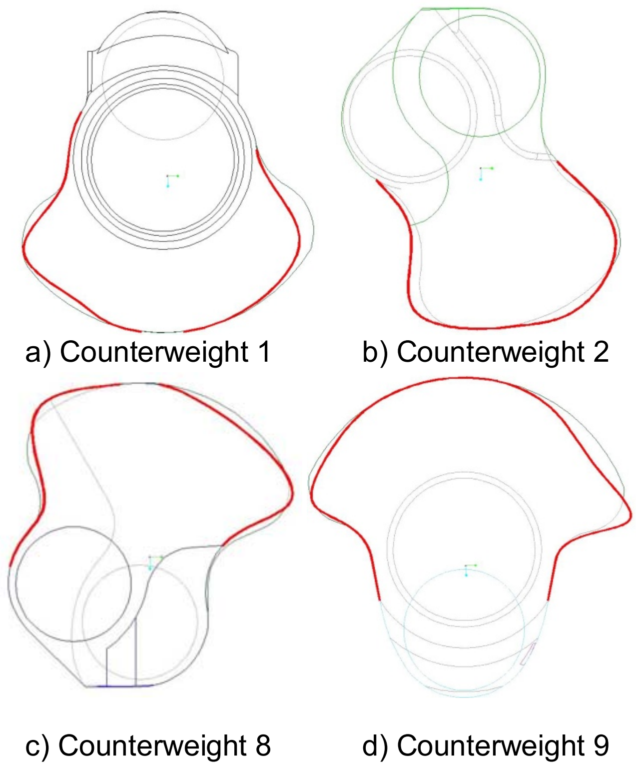


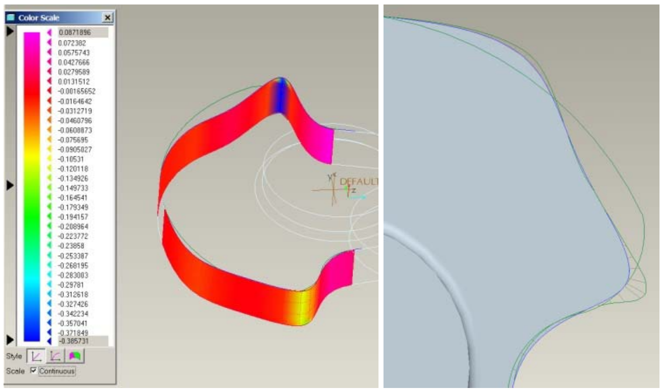

**Figure 5-16 Representations of the first attempt with the counterweight (dark line).**

An open-minded designer should recognize that the paradigm is challenged and a new concept is arising. In other words, a break with the notion that the counterweights have to be convex arc-shapes and accept the use of the splined smooth profiles with curvature changing signs. The intention of these systems, as mentioned at the introduction of the thesis, is to present challenging alternatives to the designer. Finally, these proposals are solution triggers that should inspire him/her, but, at least at this present time, they do not substitute his/her role in selecting the most suitable solutions and implementing them properly.

### 5.4.3 Results of multi-objective optimization between imbalance and CW curvature

A corresponding constraint to the optimization strategy is formulated next. An additional objective function was added: the measure of the curvature of all the splines from the profiles of counterweights. It is known fact that the curvature is the inverse of the radius of an inscribed circle of the curve. In this case, it was decided to integrate the required inscribed circles and analysis features into the geometry to extract the maximum curvature along the profiles of the four varying counterweights. The resulting objective function is the sum of all maximum curvatures, with the aim of reducing it (for instance, avoid the sharp edges by increasing radius of the inscribed circles). A new MOGA run was performed. One objective function was the crankshaft imbalance, and a second was the previously described sum of curvatures. From this new run it was possible to reduce the sharpness of the profiles, particularly from CW9, which was reduced to the shape shown in Figure 5-17. With a maximum curvature of 0.2 in the sharp edge, that is, a minimum radius of 5mm, something better than the previous shape that had a curvature in the same vertex of 0.4 (radius 2.5mm) was reached. Such a small radius in shape is not permitted in common die design rules because of the stress concentration in the die.

On the other hand, the imbalance of this individual presented a very close-to-target result, a positive development direction.



a) before c) after improvement

Figure 5-17 Curvature in CW9 profile showing an improved curvature

In the next part of this thesis, an additional evaluation will be introduced: the dynamic response of the crankshaft in order to control the first eigen-frequency, with the aim of not affecting the weight. As in this first approach, the GA will be used to produce automatically alternative crankshaft shapes for the FEM simulator program, to run the simulator, and finally to evaluate the counterweight's shapes based on the FEM output data.

### 5.4.4 Results of multi-objective optimization results between imbalance and eigen-frequency

The dynamic response of a crankshaft can be predicted based on a finite element analysis of its structure. Even when calculation of eigen-frequencies for large structures is computationally expensive, the simplification of the CAD model allows evaluations in an automated evolution process. Further JAVA programming was performed to also link the CAE software to the strategy. After an individual is assigned to the CAD geometry, an additional routine in the interface permits the finite element meshing of the CAD geometry to be performed and exported to an input file that is read by a FEM solver. The strategy allows this process to be automatic. Once the FEA result (the eigen-frequencies) is ready, it is attached to the file containing the result from the imbalance and, together, sent back to the GA to be related to the corresponding individual and let DAKOTA do the evolution of the generation.

In Figure 5-18 is shown the first generation (dots) and the last generation (squares) for optimizing the multi-objective system, with the two objectives (eigen-value and imbalance fitness, see previous section) in each axis. It can be observed in the graph that the direction of improvement of the eigen-value is upwards while imbalance fitness is towards the zero targets. The conflict of objectives can be appreciated as in the previous section. In the graph, it can also be observed that the range of the balance fitness is very broad. This is due to the very sensitive influence of the counterweights on the geometry.

On the other hand, the first mode variation is not changing very much. The perception of the influence of the geometry on the dynamic response of the crankshaft is not as significant as the balance at the level of the system involved in the analysis. Nevertheless, the direction of improvement moves to a concentrated area in the upper-left hand corner of the graph, though the balance is still not reaching the target. An individual distinguished by its position to the left was chosen as the best and the profiles of its counterweights were extracted.

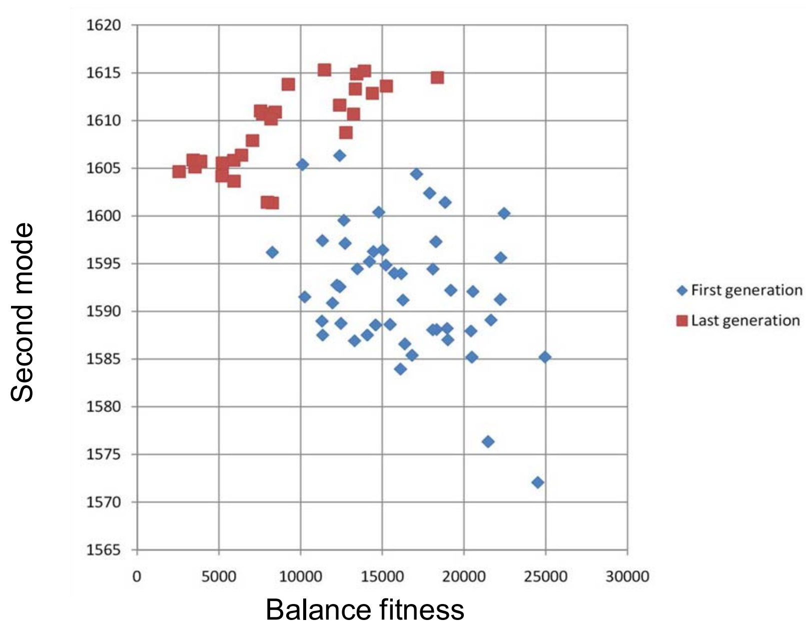



**Figure 5-18 Distribution between imbalance and second eigen-mode evaluation.**

The FEM evaluation of the eigen- frequencies of a single individual with 5000 elements takes about 200 seconds using a PC with a Pentium 4 dual processor of 3 GHz and 1 Gigabytes RAM. It is approximately a 60 hours long optimization run. This is a long time compared to the run with only geometric imbalance, so performing an inheritance evaluation of the eigen-frequency is being considered to avoid doing the FEM analysis for every individual, and instead, do an interpolation of the parent's values for a small percentage of the offspring in the population. This method is described in [87].

Nevertheless, the direction of improvement moves to a concentrated area in the upper-lefthand corner of the graph. However, the balance is still not reaching the target. An individual distinguished by trespassing the intended Pareto frontier was chosen as the best and the profiles of its counterweights were selected, see Figure 5-19.

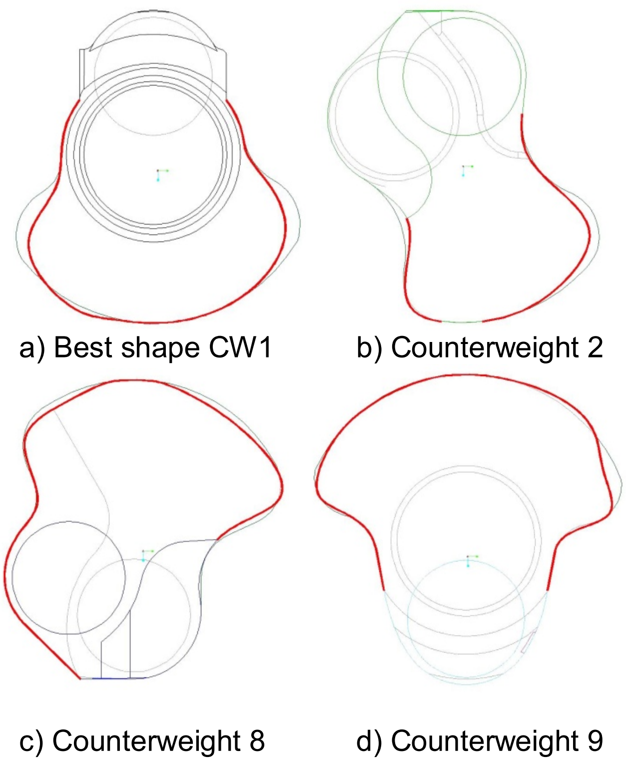



**Figure 5-19**: **Crankshaft shapes after moga evaluation (new profile in dark line)**

From the images it can be noted that the shapes of CW1 and CW2 are less wide and better shaped than in the first approach. However, CW8 and CW9 still need improvement. This profile needs to be further smoothed to allow better material flow during the forging process. The process of making smooth and less sharp a profile is a continuous refinement that is made taking into account the different objective functions. Nevertheless, the difference in the evolution is noticeable from the beginning of the single objective task, compared to Figure 5-13.

Additional to the shape of the counterweights, the corresponding imbalance graph was extracted, as can be seen in Figure 5-20. From this graph it is inferred that, even when the shapes of CW1 and CW2 look better, the balance is not as close to the target as the first approach was. But further iterations will be performed to improve both the imbalance and the curvature of CW8 and CW9.

The workflow presented offers all necessary features for optimizing the parts considered and/or does robustness and reliability evaluations. In the case discussed in this thesis, both workflows have been used for evaluating the robustness of the relevant eigen-frequencies against variations of the geometry of the part. In a first step, the sensitivity of the stresses has been evaluated in a structured DOE. This led to the identification of the most important measures. A response surface has been modeled, describing the dependencies between the stresses and the identified measures. After the surface was verified, a multi-criteria-optimization of geometric design parameters was performed with the objective functions “Eigen-frequencies” and “Balance”.

An automated generation of FE meshes and FE analyses of components can be executed by the functional range of the current CAD software packages and additional scripts that are comparatively easy to handle. By using established solvers like MSC.Nastran and automation of the CAE processes, it is also possible to use the CAD software packages. When joining CAD and FEM, there are further advantages: The application of established solvers enables a direct comparison of analysis results with classically generated FE models. The applied boundary conditions are revisable and controllable. The processes presented offer all the features for optimization and/or robustness and reliability evaluations of a component. Restrictions in automation are rather natural and are applicable to alternative processes such as mesh-morphing.

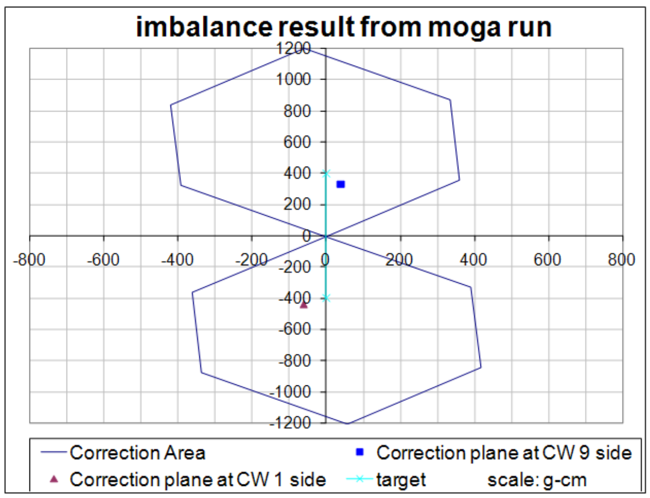



Figure 5-20 Imbalance graph of best individual from second approach

## 5.5 Chapter conclusions

During the development of the chapter, it was possible to deploy the implementation of the tasks that are built upon the optimization/innovation strategy described in previous chapter. It was described the construction of the CAD model and how the counterweights of the crankshaft were transformed from simple arc lines to a spline curve, one of the key elements of the strategy. The chapter describes how the imbalance formulas, presented in section 2.2.2 were introduced to the model. Elements also described were the curvature and the eigen-frequencies analysis. The control interfaces developed in Java language was introduced. Such Java interface allows the CAD software and the Finite Element Analysis software to communicate and interact in order to coordinate the results of imbalance and the eigen-frequency value. The interactions permits the CAD software to generate a FEM mesh, with corresponding boundary conditions, and send it to the solver for solution.

In the chapter, the results of the optimization tasks are presented, together with an analysis and comments about the shapes obtained. The repeatability of the results is a factor of possibility, because of the randomness of genetic algorithms. Nevertheless, during the first task that consisted in a single objective of imbalance, two different algorithms were used an compared: vgGA and DAKOTA GA. Both algorithms presented a similar behavior in terms of convergence during the generations. However, the convergence is not towards zero, as originally expected, which means is not reaching the optimum. Another interesting result was the shape of the counterweight that resulted during this task: the appearance of notches in the profile near the joint with the mains. It is a condition that can be read as if the mass were separating from the crankshaft, a condition that has been implemented in crankshafts from other engines. Then it was decided to perform a multi-objective optimization using the imbalance from both sides of the crankshaft (both correction planes) as two fitness functions. For the run it was used a DAKOTA moga and the results, deployed in graphical format, reveals a movement of the individuals, from a evenly distributed dispersion, towards a value of zero in the last generation. The shapes of the counterweights resulted in a form that, even when improves, still has sharp edges. So further improvements were needed using curvature analysis to reduce the sharp edges in the profiles.

The last optimization task of the chapter was a multi-objective optimization with imbalance as an objective function and the second eigen-frequency as the other objective function. The results were presented in graphical format and the direction of improvement that was appreciated goes from a wide dispersion in the first generation towards the upper left corner of the graph. It is worth to mention that the shapes of the counterweight resulted in a more smooth and less sharp edges, also the width of the profiles are smaller than the original designs. This result needs to be validated and a proof of concept will be presented using fem simulation of the forging process in the next chapter.

# Chapter 6 The Optimization / Innovation Approach

Even when the design of the counterweights is not totally finished, a designer should be able to recognize that the paradigm is challenged and a new concept can be derived, in other words, dropping the notion that the counterweights have to be arc-shaped and allowing the splined but smooth profiles. The intention of these systems is, not only to create a totally automatic optimized solution, but also to provide the designer with challenging alternatives. In reality, these proposals are solution triggers that inspire him/her, but they do not substitute his/her role in selecting the most suitable solutions and implementing them properly.

This chapter presents two aspects that need to be further studied but general basics are stated. One is the extension of the optimization task to include the simulation of forging by FEM. The proof of concept using simulation of forging is an indicator of the advantage of the approach and it is shown. A second part of the chapter deals about how the topological development of products can be enhanced with new paradigms. Particularly the integration opportunities between genetic algorithms and TRIZ

## 6.1 Simulation of forging vs. curvature of counterweights

The next step of the strategy consists of correlating the curvature of the counterweights to the forge-ability of the crankshaft. A correlation between curvature and metal flow can help reduce computational time. Identification of potential curvature parameters to be evaluated is necessary to find correlation with metal flow in forging mechanics. The intention is to reduce the absolute value to a certain limit, defined by the forging simulation results.

There are two ways to make this correlation possible. The first involves the automatic manipulation of forging software using a Java interface. As in the solution of the eigen-frequencies FEM solver, there is FEM-based forging simulation software with script capabilities, and one possibility is to include the simulation of the forging into the optimization loop. The drawback of this option is the time required to complete the evaluation of an individual. This may involve extracting the profile of the counterweight from the CAD software to the FEM software; preprocess the forging conditions including the billet; running the simulation, and extracting the results of the forging's material filling.

The second possibility is to create a table that correlates the curvature of the counterweights' profile, with some results from simulations of the forging process. This option allows every individual from the optimization loop that has a curvature evaluation to be assigned with a value of forge-ability previously defined and tabulated. This way a forging simulation does not have to be performed during the optimization, but can be done in advance; and the time to complete each evaluation is much less. The drawback of this option resides in the possibility that the results have to be interpolated to individuals that do not match any value included in the predefined table.

Both previously mentioned possibilities imply the importance of defining which forging parameter should be used as objective function. The most convenient is the filling of the cavities of the forging dies. Given that the root cause of the forging's imbalance influence is the under filling condition of crankshafts; as stated in the introduction, it is important to include it as a main optimization direction. An additional strategy can be considered in the thesis due to the particular nature of the phenomena analyzed: fitness inheritance.

A typical genetic algorithm (GA) operates by using a population of potential solutions to create a new population via selection and mating. Then, the GA eliminates the old population (of parent solutions), and evaluates the new population (of offspring solutions). Thus, the fitness values of offspring are related to those of their parents only through genetic inheritance [87]. In the natural world, a child's legacy can be more complex. In species that conduct parental nurturing, parents effectively "pass on" some of their fitness to their children in a variety of ways. A child's survivability is enhanced by having successful, protective parents. In addition, parents can pass on their fitness-related knowledge and experience to their children through teaching. Moreover, a parent can pass on fitness through inherited wealth. In some optimization problems, it is difficult to apply a GA because the cost of determining fitness values for an entire population is prohibitive. If obtaining the fitness of a single individual involves a complex calculation, like a finite element model or a lengthy simulation, a population-based approach may be impractical. Certainly, a low-cost way for an individual to survive is to obtain fitness the old-fashioned way: inherit it.

The use of the JAVA interface allowed the genetic algorithm to be integrated into the CAD software, as described before, and the further integration with the CAE software, as described in the second part. The development of a Pareto frontier from the first analysis and a Pareto with the additional eigen-value objective were possible. The following important conclusions were extracted:

- It is necessary to prevent the development of sharp edges in the shape of the counterweight by adding extra control of the curvature and minimal radii.
- Simulation of the forging process is a required next step for definition of the relationship between good shape-curvature and manufacturability. This is very important when a proposed design outside the initial shape restrictions needs to be justified to avoid affecting formability.

Subsequent activities include the definition of additional fitness functions not only in CAD but also in CAE simulation (forging simulation) for the control of “strange” shapes. Objective functions and restrictions will be added with the use of forging simulation and stress analysis during variations in geometry. The aim is to integrate different systems that run totally or partially automatically.

## 6.2 Proof of Concept by Forging Simulation of New Counterweight Profile

The next step was to perform a proof of concept of the new profile of the counterweights using splines. A simulation of the original profile of CW2 with the following characteristics was performed:

- The preform is a circular bar with a 117.7 mm diameter
- Temperature of the bar: 900°C

- Temperature of the dies: 400°C
- Specifications of the mechanical press: stroke: 520mm, velocity: 0.533 strokes /sec
- Current stroke at the moment of deforming: -468.24mm

The initial conditions of the simulation can be seen in the Figure 6-1. The color mapping is the contact distance between the dies and the work piece.

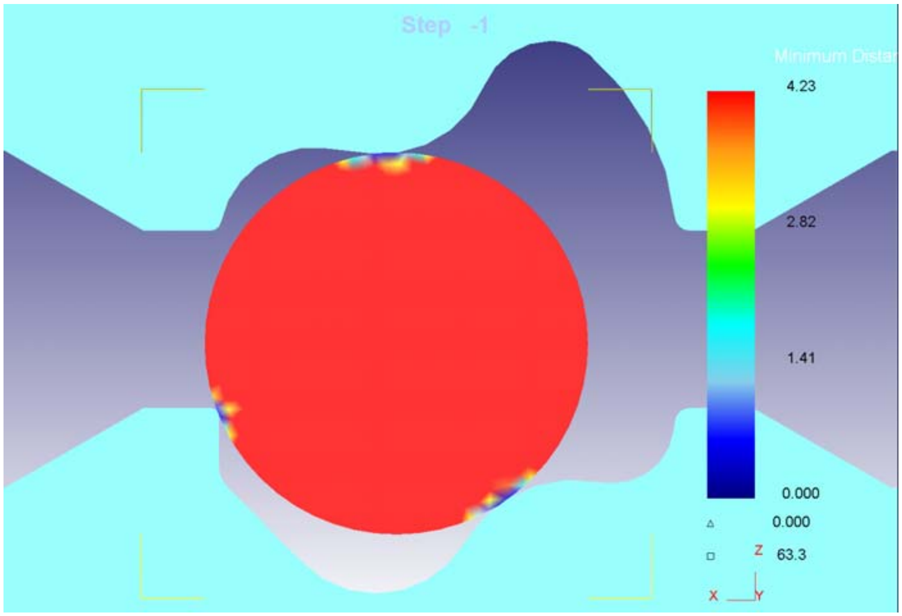



Figure 6-1 Initial conditions of the forging simulation.

The Figure 6-2 shows the final conditions of the simulation of the original profile. It is possible to see that the material didn't fill out all the die cavity.

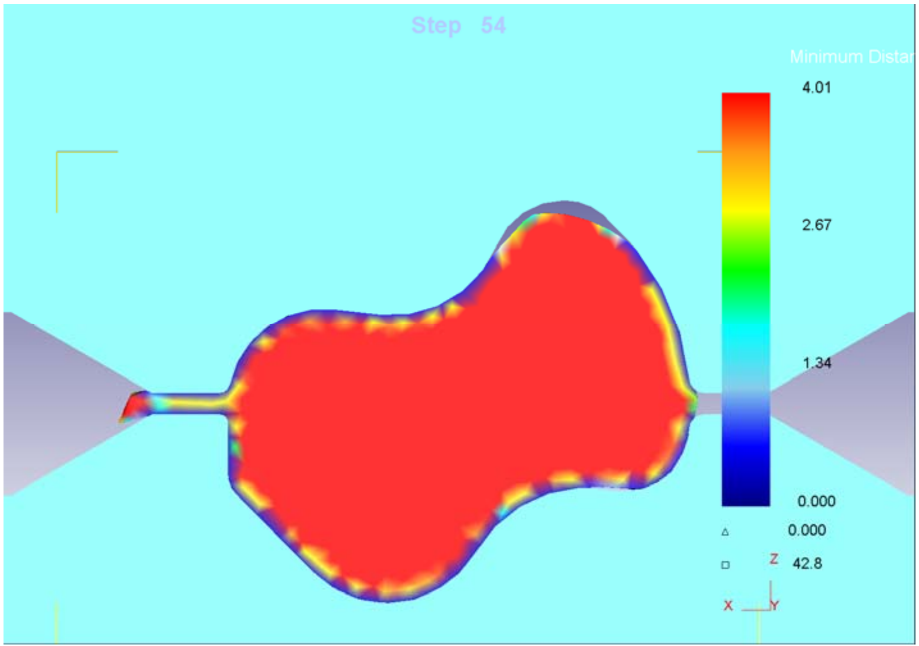



Figure 6-2 Final conditions of simulation with original profile.

The Figure 6-3 is a simulation of the forging using the profile concept resulting from the use of the development method of this research. It can be seen that the die is better filled with material.

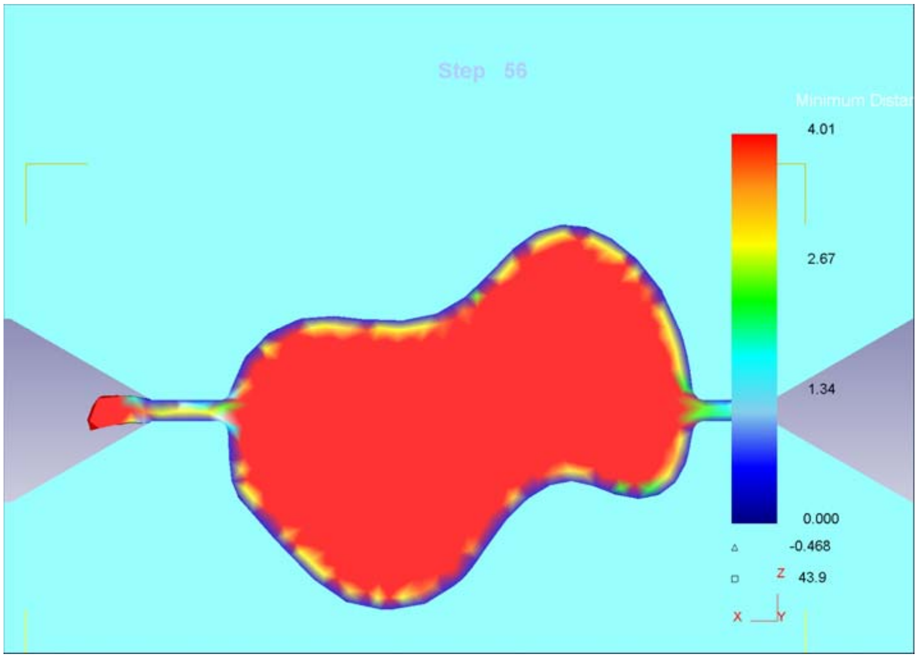



Figure 6-3 Final conditions using a splined profile

## 6.3 From evolutionary design to computer-aided innovation

The aims of this research program are to extend the “detailed design” capabilities of computers, and to merge them with “conceptual” activities during the product development process. Within this context, the use of genetic algorithms is put forward as a mechanism for mediating conflicts. They are able to reveal the conflict that may exist between parameters, shown by the different fitness functions. When a designer is solving a multi-objective optimization problem, he or she tries to find a vector of decision variables which satisfies the constraints and optimizes the objective functions. These functions form a mathematical description of the performance parameters, which are usually in conflict with each other. Hence, the term “optimize” means to find a solution that will give the values of all the objective functions an “acceptable trade off” to the designer [62]. To deploy the concepts explained, the Pareto diagram is used (mainly in multi-objective optimization processes). Such a diagram shows a boundary that divides the region of feasible solutions from the region where restrictions or physical constraints prevent the solutions from acquiring certain values.

An airplane is designed to fly a range of missions under a variety of externally imposed conditions. The combination of missions and external conditions determines a performance envelope in which the aircraft is expected to perform well. For example, an airplane might be designed to fly in a range of temperature, pressure, and velocity conditions, with specified degrees of turbulence, payload, maximum crosswind at landing, and so forth. As long as it operates within the specified envelope, and as long as the aircraft has been properly designed, the aircraft is expected to perform satisfactorily. When the boundaries of either the mission or external conditions are surpassed- when the envelope is pushed -or when the aircraft is improperly designed the increased likelihood of less-than-satisfactory performance or failure can be anticipated [88].

Performance improvement of a product is commonly obtained through quantitative changes of a parametric design (optimization) and, once the improvement through optimization reaches its limit, new searches must be carried out through qualitative changes and paradigm shifts (innovation). Certain innovations can be achieved through methodologies based on optimization processes that focus on functionality and performance where not only design parameters, but also shapes and topologies can be changed. In other words, it may be stated that product innovation may also be implicit as “optimization not restricted to parametric variation” but extended to ‘constrained concept variations’ [89]. As has been shown in several case studies, an extended parametric optimization is achieved by adding shape and topology as possible search directions. This extended optimization has been achieved by allowing shape variation through substitution of the original shapes based on straight lines and arcs with spline curves, or by eliminating finite elements from a meshed structure to reduce it to a new shape or by considering predefined alternative shapes. An extended shape variation in tree-structured CAD systems represents a possible enhancement to allow an "automatic" control of the shape variation not only when shapes are represented by parametric curves but also when represented as 3D shapes in 3D parametric CAD packages in which they are built from constrained parametric profiles and Boolean operations. This represents a further step toward adding "concept variation" to optimization procedures.

### 6.3.1 Multi-objective optimization, conflicts and the ideal solution

Solving optimization problems with multiple, often conflicting objectives is generally a very difficult goal. Although single-objective optimization problems may have a unique optimal solution, multi-objective problems, as a rule, present an innumerable set of solutions. When evaluated, the solutions produce vectors with components that represent trade-offs. TRIZ deals with the concept of the Ideal at the moment of resolving trade-offs [90]. According to TRIZ the ideal final result is a concept that represents when the function is performed but the systems that carries it doesn’t exist in reality, a paradigm that serves as mental inertia breaker. However, the concept of the ideal point has also been taken up in other studies, as described by Coello [91]. These studies explain that attempts have been made to develop a methodology emulating the criteria that drive a human being to adopt a certain solution named the ideal point [62]. However, there is a set of optimal solutions lying on a line that prevents the functions from reaching the “ideal” at the same time. This notion is represented in a Pareto diagram (see Figure 6-4. Given the constraints in the solution space, the ideal point lies in the unfeasible region. It is the place where the objective functions may lie if the system doesn't exist, as explained before.

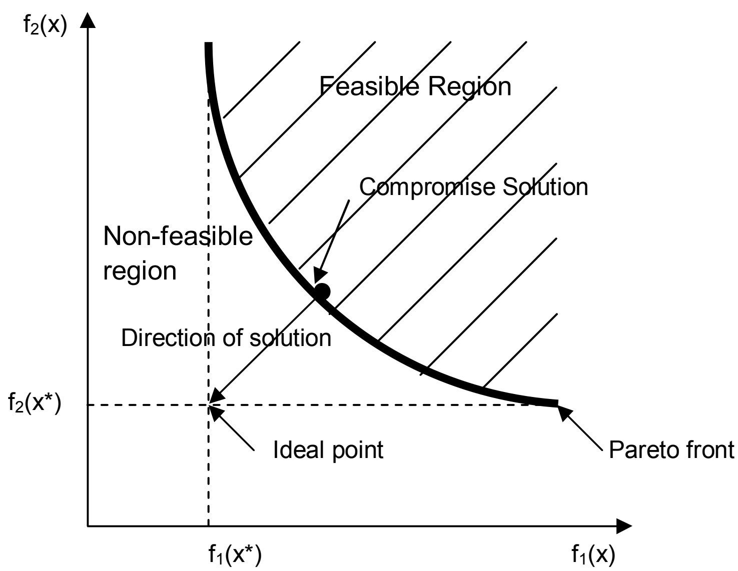



**Figure 6-4 Pareto diagram and the concept of Ideal.**

To solve this, a decision must be made by exploring the achievable limits along each particular attribute of importance (parameter) in order to find an ideal alternative. Some authors propose performing a set of mono-objective optimization tasks to reveal conflicts [75]. TRIZ proposes the use of the inventive principles. The idea is to direct algorithms as "conceptual machines" and let computer scientists hesitant to move forward without exact models precisely describing their situation to collaborate with engineers, who have a more practical orientation. How many fitness functions are enough? How many objectives are generally required to adequately capture the essential characteristics of a multi-objective problem? Can all relevant characteristics be captured? "Where do I begin?" We cannot specify an "all purpose" multi-objective technique. However, we can suggest algorithms that appear appropriate as a starting point.

### 6.3.2 TRIZ and cataclysmic operators

Traditional TRIZ theory encourages the attainment of an Ideal Final Result and offers tools for identifying the underlying technical and physical contradictions in a technological system. TRIZ general solutions (i.e. inventive/separation principles, Standard Solutions, etc.) are proposed to overcome the conflict and let the product evolve according to the "laws of technical evolution" [92]. This is a natural convergence direction to merge evolutionary design (based on laws of biological evolution) with TRIZ (based on laws of technical evolution) within a computer framework suitable for computer-aided innovation. Other authors also apply the TRIZ methodology to the structural product optimization [75]. Still others are using GAs to generate creative forms and shapes [9], but the main difference with them lies in the merging of TRIZ inventive principles with genetic algorithms. In other words, genetic algorithms controlling splined shapes are the core interface to apply innovation operators from TRIZ to the development of a product in a CAD&CAE environment.

Technical change is an evolutionary process; technology keeps on producing entities superior to those in existence earlier. Technological change and innovation processes are thus 'evolutionary' because of their characteristics of non-optimality and of an open-ended and path-dependent process.

Altshuller [90] states that:

- Technological systems evolve not randomly but according to objective laws of evolution. These laws do not depend on humans. They should be observed, formulated and used in order to develop efficient methods of problem solving.
- Technological systems evolve not randomly but they have to overcome contradictions. To get breakthrough idea we should find a way to overcome contradictions.
- Each specific problem must be solved in accordance with the restrictions of the specific problem situation, with peculiarities in each specific case, and cannot be given a general solution. A robust solution is a solution that involves as few new resources as possible.

Some authors have tried to link system parameters and laws of engineering system evolution in what is called an evolution hypothesis [93], in research seeking a logical use of these laws of technical evolution, within the choice of the appropriate pair of parameters in conflict. Those previous efforts are converging in the extended use of genetic algorithms.

In the research on which the present thesis is based, the aim is to extend the paradigm of multi-objective optimization to reach the ideal point, in this case the "Ideal Final Result". GAs can be enriched by taking advantage of the inventive principles, and extending the operators beyond basic "mutation" and "crossover". Additionally, it is possible to have new operators or "agents" capable of modifying the way the algorithms operate on the CAD geometry [89]. In this way, CAD systems can develop new configurations and alternative modifications to the geometry, and supply the designer with a set of alternative modifications, defined automatically on the basis of the selected principles, which may themselves be applied on the basis of the concept of cataclysmic mutations [94]. Mutations following a pattern similar to a "cataclysm" are now being studied in the area of Evolutionary Algorithms as tools for finding innovations [95][96]. The extension of the solution can reach a level of detail that leads to two possibilities: either automatic development of solutions by the genetic algorithm with or without the use of cataclysmic operators or, on a more detailed level, inspired by these suggestions, the designer can select the most suitable solution and implement it.

In many cases, the use of TRIZ can facilitate the design optimization process, giving a qualitative lead to what in the model may need to be changed. These leads can be then validated via FEA. To design a product that meets a set of performance criteria, it is often necessary to go through several iterations of computer modeling (e.g., using finite element analysis) and then make design changes. The optimization process requires determining sensitivity parameters of the FEA model, which leads to complicated derivations and then time consuming calculations. The number of calculations can be reduced if there is some understanding of how the optimum design should look, based on a simplified model, or intuition. Unfortunately, the tasks like those described above are often counter-intuitive, and developing an adequate simplified model can be a challenge in itself.

A researcher who intends to combine in the optimization process highly structured and well established numerical procedures, such as FEA and sensitivity analysis, on the one hand, and TRIZ recommendations, on the other, has to be able to overcome psychological challenges as he or she tries to interface the two approaches. The TRIZ statements need to be rephrased and made more custom tailored for the application in question. They also need to be interpreted in terms of those variables that can be changed in the problem. The TRIZ recommendations need then to be verified through direct modeling. However, a proper application of TRIZ to the highly intensive numerical procedures can make a difference between getting the solution or no solution. Moreover, a systematic decision-making process for accomplishing product innovation is necessary, in accordance with the target quality, i.e. the target values and relative weights of the relevant quality characteristics of the product to be developed. An approach of a decision making process with the use of TRIZ and using a die-casting machine as an illustrative example is described in [97]

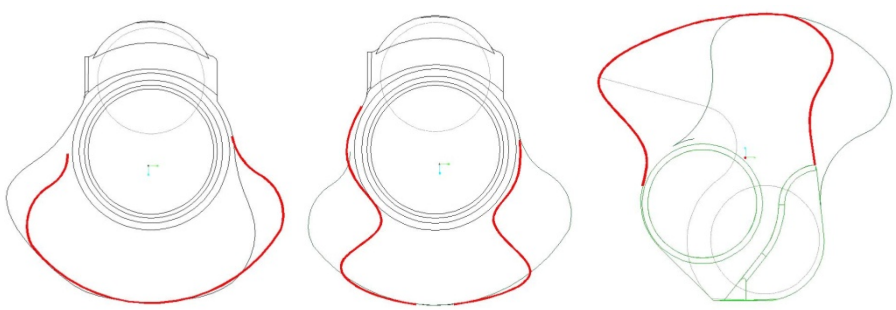
Inventive principles suggest a series of recommendations to change the direction in which solutions are sought. The recommendations are based on the fact that they have provided solutions in previous analogous technical problems. These recommendations can be regarded as a knowledge database, which can be used to feed the cataclysmic transformation of genotypes in the course of evolution for optimization, allowing them to override the barriers of contradictions or constraints. The TRIZ inventive principles with a geometric interpretation can be added to form the extended cataclysmic operators. The level of impact from the different operators can vary from a slow, steady accumulation of changes (the way an optimization algorithm normally performs), to a sudden disturbance in the nature of the system (a cataclysm). The most important effect is the creation of a jump in the phase transition. SeeTable 4.

**Table 4 Comparison of genetic and inventive principles**

| GA operators | TRIZ principles | Representation |
|---|---|---|
| Crossover, simple | Segmentation, combination | Divide two genotypes and combine alternate parts |
| | Asymmetry | Break symmetrical genotypes |

| | | |
|---|---|---|
| | Merging | Join genotypes (similar or not) |
| Crossover, nesting | Nesting | Place part of a genotype inside another |
| Inversion | Inversion | Turn around a genotype |
| | Another dimension | Create genotypes from different parameters |
| | Homogeneity | Make a genotype homogeneous |
| | Discarding and recovering | Break down and rebuild genotypes |
| | Spheroidality | Join the beginning to the end to form a circle |
| Mutation | | Change a gen in a genotype |
| Selection | Feedback | Return fittest genotypes |
| | Copying | Take a copy of fittest genotypes |

Some examples of how the agents may be applied to the case study can be seen in Figure 6-5.



a) Asymmetry b) Separation f) Inversion

**Figure 6-5 Graphic representation of inventive "agents" (dark lines).**

More suggestions can be enriched by means of guidelines provided by the inventive principles that can be associated with the genetic operators. As result, the algorithm should be capable of applying agents according to the conflict being faced.

Directly interpreting TRIZ principles in a geometric context could lead to a wide number of possibilities. During an optimization process with genetic algorithms, any of these TRIZ interpretations could provide valid options during the process. It is important to note that some principles are related to the function of the product, so they can be applied more directly. The absence of direct functional information about the overall characteristics of the design, during the optimization process, could reduce the possibilities of a valid solution. On complex geometries or time consuming optimizations where manufacturing constraints play an important role, more restricted TRIZ interpretations could be the best suited [27]. TRIZ concepts could prove useful in suggesting modification possibilities based on some of the TRIZ innovation principles. Such principles can be identified by generalizing existing design contradictions in the given part to obtain suggestions from a predefined contradiction table. These suggestions should enunciate the modification principle along with a design-oriented example depicted by CAD models.

Other inventive principles are of a rather topological nature and therefore may be implemented in CAD systems' assembly modules. In other cases the principles are of a mechanical or physical nature, which also involves the effect of time and other physical parameters such as velocity, force, acceleration, temperature, etc. and may be implemented using multi-body systems. Normally a graphic description along with a picture or drawing depicting the given suggestion is also provided. With these examples, the user would have a much better idea about where and how the shape modification process should be focused. However, the designer has to implement the required modifications on his or her own by editing step by step the actual shapes and topologies based on how he or she understands the recommendation. This is commonly a time-consuming task that avoids the search for better solutions.

In this study, the results from the first attempts show that the imbalances from both sides of the crankshaft are in conflict with each other. A resolution of these conflicts are then sought using “innovation agents”. Further development of the algorithms can only be achieved by its integration with innovation methods. The resulting systems are of a parametric shape and topology innovative configuration. Some features need to be added to the system for it to work in an "out of the paradigm" way that will lead to solutions not considered before. To give a visual impression of the way the algorithm performs, some of the counterweights are shown in Figure 6-6.

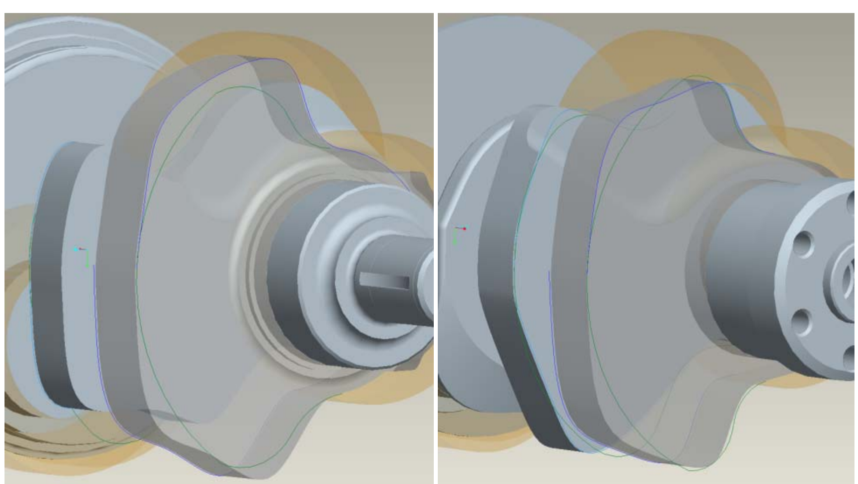


**Figure 6-6 Representations of the crankshaft´s counterweights (transparent)**

The shapes present some notches that are not suitable for forging, but the direction of the solution is cataclysmic. An open minded designer should be able to recognize that the paradigm is challenged and a new concept can be derived. This is the intention of these systems, as mentioned at the beginning of the thesis, presenting the designer challenging alternatives. Finally, these proposals are solution triggers that inspire him or her, but they are not substituting his or her role in selecting the most suitable solutions and implementing them properly.

Even when the innovation principles cannot be codified in a genetic structure, it is possible to treat them as a concept called “meme”. In the book “The Selfish Gene” [98], Dawkins speculated that we may now be on the threshold of a new kind of genetic takeover. DNA replicators built 'survival machines' for themselves - the bodies of living organisms including ourselves. As part of their equipment, bodies evolved onboard computer-brains. Brains evolved the capacity to communicate with other brains by means of language and cultural traditions. However, the new milieu of cultural tradition opens up new possibilities for self-replicating entities. The new replicators are not DNA and they are not clay crystals. They are patterns of information that can thrive only in brains or the artificially manufactured products of brains - books, computers, and so on. But, given that brains, books and computers exist, these new replicators, which Dawkins called memes to distinguish them from genes, can propagate themselves from brain to brain, from brain to book, from book to brain, from brain to computer, from computer to computer. As they propagate they can change - mutate. In addition, perhaps 'mutant' memes can exert the kinds of influence that Dawkins is calling 'replicator power'. This means any kind of influence affecting their own likelihood of being propagated. Evolution under the influence of the new replicators - mimic evolution - is in its infancy. Additionally, during this process new ideas can arise in what has been called “emergence” [99]. An explanation is given of some frameworks for both representation and process for a computational model of creative design that are based on design prototypes and on an evolutionary model. This phenomenon of emergence is deemed to play an important role in design [100]. Computer-based design assistants can and should support this human perceptual ability, using pattern recognition to anticipate human designers’ perception of emergent shapes and supporting the subsequent manipulation of and reasoning with these shapes as part of the design.

## 6.4 3D Model Repositories as Knowledge Databases

Recently the appearance in the market of so-called innovation software presents a combination of TRIZ methods and instant access to knowledge databases. Those databases consist of patent databases and knowledge documents. It would be useful to exploit the valuable knowledge embedded in repositories of digital models where search engines are capable of expressing high-level and advanced queries, which can effectively support the re-use of CAD models. The retrieval mechanism should be able to return not only global similarity measures among objects, but it should also be coupled with methods for the automatic identification of similar sub-parts in the query and retrieved models, possibly highlighting automatically the object differences [101]. Technological improvements related to object acquisition, visualization and modeling, have caused a considerable growth of the number of 3D models in digital form, which are now available in large databases of shapes, ranging from unstructured repositories. This is especially crucial for CAD model databases which contain designed shapes: it has been estimated, indeed, that a large part of the design activity is based on the reuse of previous knowledge to address new design problems. Therefore, a retrieval system closer to the human perception of similarity should be able to assess similarity to identify the common parts (global and partial matching). Structure-based shape description is particularly important for engineering applications. The use of structural descriptors is an intermediate step between the geometry and the semantics of the object, since it characterizes features of the object that are relevant with respect to the application context or the design activities, like holes, slots and handles.

To perform content-based indexing and retrieval of 3D objects, each model must be converted into some collection of features [102]. Previous research on model matching and retrieval has drawn on feature definitions from mechanical design, computer graphics and computer vision literature. Many of these feature-based techniques ultimately use vertex labeled graphs, whose nodes represent 3D features (or their abstractions) and whose edges represent spatial relations or constraints, between the features. Retrieval and matching is done using some variation of graph matching to assign a numerical value describing the distance between two models. It is common in engineering communities for the term feature to be used to refer to machining features (i.e., holes, pockets, slots) or other local geometric or topological characteristics of interest, depending on the domain (i.e., assembly surfaces, molding lines, etc.). In the context of this research, feature will be used as an intrinsic property of the 3D shape which may encompass local geometry and topology. Feature-based descriptions of models also vary by system. Hence, CAD search tools that can perform semantically effective searches using “the lowest common denominator” (e.g., shape) representation are widely applicable.

Unlike shape models, for which only approximate geometry and topology is available, solid models produced by CAD systems are represented by precise boundary representations. When comparing solid models of 3D CAD data, there are two basic types of approaches for content-based matching and retrieval: (1) feature-based techniques and (2) shape-based techniques.

- Feature-based approaches. Historically, Group Technology (GT) coding was the way to index parts and part families. This facilitated process planning and cell-based manufacturing by imposing a classification scheme (a human-assigned alphanumeric string) to individual machined parts. While there have been a number of attempts to automate the generation of GT codes, transition to commercial practice has been limited.
- Shape-based approaches. Comparing CAD models based on their boundary representations can be difficult due to variability in the underlying feature-based representations. Additional complications are created by differences among the boundary representations used by systems (i.e., some may use all NURBS; some may use a mix of surface types, etc.).

Feature-based modeling and tree structures of part geometry of CAD systems are suited for the linking of shape conception with the functional requirements. But little effort has been made in integrating the functional requirements with the CAD system structure. As a result, CAD environments usually ignore functionality and focus on the geometrical, detailed design process.

This situation enhances the psychological inertia of designers and their reluctance to introduce concept changes because it is necessary to perform difficult operations to introduce non parametric modifications to the parts, sub-assemblies and assemblies.

The evolution trends of CAD systems has been towards systems integration [27]. The inclusion of CAM, CAE, PLM and others has been taking place at different integration levels. However, another evolution trend could be the inclusion of knowledge modules in CAD systems. These knowledge modules allow the creation of functional descriptions of the product at abstract levels before the part geometry is defined. The functional relationships between components is made by different boxes and arrows that determine the interactions between each design component and show whether they have positive or negative effects on the principal function.

Those tools for functional decomposition and interrelations in a given design, could be more easily integrated with an entirely new CAD-integrated application that establishes such interactions in a parameter-free environment. That environment could prove useful in relating TRIZ innovative principles with modification suggestions for a given design, and the designer would not be restricted to the current geometry but would be free to manipulate the CAD geometry. Such a tool should enhance the standard CAD hierarchical specifications tree with information about the functional relations existing between features, and could enable the possibility of generating multiple functional variants be integrated with the shape and topology creation process. In a new design paradigm, such structures could prepare the way to generate shape and topological variations in CAD systems, which can contribute to stimulating the designer's imagination to generate more design variants by reducing the psychological inertia.

The difference between an assembly and a part in a 3D-CAD system is determined through the type of relationships established inside the model tree. Among features, the relationships for building a part are Boolean type operators, such as join, cut, merge and so on. At assembly level the relationships among neighbors are of type mate, insert, align, etc. Functional similitude between two features/parts/assemblies may be derived from similar declared functions as human identified or perceived actions in order to distinguish them from the behavior concept, which is identified when simulating the performance in a CAE environment. A behavior is, therefore a simulator-identified action or action value and does not require human intervention (object action-reaction-object-result as basis for functional behavior analysis). The functional structure may be introduced in the geometric tree of parts and/or assemblies, by the human user, for analysis purposes. This is the type of analysis that is performed today with existing TRIZ software. In the case of a CAD system with functional information, the analysis performed may lead to the recommendation to use some of the "genetic operators", which could be applied to the related features/parts/assemblies for visualization purposes. That means the geometric structure is varied with the aim of suggesting to the CAD model, possibilities for overcoming the problems formulated through the functional structure. A further step occurs when the changes introduced (from changing operators or from crossbreeding operations) are analyzed by CAE tools (FEM, CFD, MBS) for analyzing its behavioral structure (stress and levels, fluid behavior, accelerations, velocities, etc) in search of "optimal solutions"

The design research community has spent much of its effort in recent years developing computer-supported design systems [103]. Generative design systems - systems for specifying, generating and exploring designs spaces and alternatives - have been proposed and studied as a topic of design research for many years. However, systems to date are just beginning to have an impact in practice. Designers have authority to decide what kind of sketch shapes, components and layout are selected. Through the interaction between the designers and the system, it is possible to produce new designs that the designer alone could not have easily created.

## 6.5 Chapter conclusions

The chapter started with some considerations on the simulation of forging as a means to provide proof of concept for the shapes developed in the previous chapter. In addition, some considerations relative to the correlation between curvature of the counterweights and the forge ability of the crankshaft are explained. The proof of concept consisted on simulating the forging process with the original profile of the crankshaft, and comparing the results against the new profile, obtained according to chapter 5. The simulations that an improvement in terms of cavity filling is gained and that promising results are possible to obtain.

Next, a sequence of reasoning and dissertation on the transition from evolutionary approaches to the emerging discipline on computer-aided innovation followed. It is explained how the present research aims to extend the capabilities of computers into a support for conceptual design activities during product development. The key elements that are having significance on the process are the multi-objective optimization, the conflicts and the ideal solution. They relate the TRIZ methodology to the Pareto diagram, where the ideal solution concept can be represented. Traditional TRIZ tools encourage the attainment of Ideal Final Result and their general solutions help to overcome conflicts while the system evolves. The laws of technical evolution converge to merge with the laws of biological evolution, from where the genetic algorithms come. Some coincidences are found between the inventive principles and the genetic algorithm operators, which lead to think on new operators form TRIZ that can be used during an optimization and provide novel solutions.

Even when the computers are not generating creativity by itself, the observance of the designer on the shapes and forms generated can influence his thinking. Ideas in the form of “memes” can trespass the barriers of algorithm codification and emerge in the mind of individuals. On the other side, the so-called innovation software that benefit from TRIZ and provides access to knowledge databases, could exploit the existence of CAD models organized in the form of accessible information, classified according to shape and topology.

# Chapter 7 Summary and Conclusions

The usefulness of computers in product design ranges from data management, drawing, analysis and simulation, to numerous other applications. Computer is extending from modeling activities and optimization tasks to simulating the performance of a product. They are programmed to do what humans intend and are capable of performing far beyond the abilities of the most skilled designer. This allows the engineering designer to concentrate on activities related to analysis, prototype testing, creativity and innovation.

Nowadays, however, designers are provided with software tools that are suited for the quality paradigm of the 80´s and 90´s, and not always adapted to satisfy contemporary innovation requirements, which are increasingly more focused on optimizing the creative solutions that, in theory, lie in a space with infinite design variants. The automation of conceptual design processes in engineering is a desirable characteristic that has been pursued many times. Automation allows designers to focus their efforts on their irreplaceable role in engineering design to improve design performance. One of these efforts has been the use of knowledge-based system techniques to develop a computational tool for creative conceptual design [104].

Methods for structural and topological optimization, based on evolutionary algorithms, are used to obtain optimal geometric solutions. They evolve into configurations that minimize the cost of trial and error. Design knowledge comprises knowledge about the structure of a product and the design process [105]. The so-called control knowledge about the design process determines in which sequence design actions should be carried out in order to minimize trial and error. Although the field of evolutionary design is showing some impressive results, computers are not fully autonomous. People are required to work out which function the design should perform, and how a computer should be applied to the problem. Evolutionary design can present characteristics that add value to the product, even by chance. If the creations of design concepts evolve by generating novel designs, e.g. novel product shapes that achieve higher performance, this can be interpreted as “being creative”. So it can be said that computers can display “creative” behaviour.

The development of a new category of tools known as computer aided innovation (CAI) is growing as a response to a development of methods and tools that are used to boost creative solutions. These new tools stand out from the current CAD/CAE/CAM tools because they challenge the previous standards. The goal of these CAI tools is to assist designers and process developers in their creative stage. Functional analysis and synthesis of the generated system is accomplished with the computer aided innovation (CAI) systems, which provide computer support in the area of innovative task formulation and solution.

Computer-aided innovation uses software tools to provide extended support for the creative part of the design process. Software tools can use knowledge from designers to generate new solutions, based on many separate ideas and suggest entirely new design concepts. With this support the designer can improve the performance of his or her concepts, letting computers take part not only in generating variants, but also in making judgments, by simulation, of these variants. Thus, a designer can explore numerous creative solutions to problems (overcoming 'design fixation' or the limitations of conventional wisdom) by generating alternative solutions.

The main problem for a "computer-aided inventing" algorithm resides in the fact that the possible concept variations of any product or process are infinite even inside of constrained spaces. The "computer-aided optimization" concept is based on techniques that reduce the search space by "sensing" the effect of variations of a reduced number of parameters involved. This means that an inherent contradiction is widening of the search space to non-parametric variations as the universe of possible solutions increases enormously to the extent that no computer-aided methods are available for thoroughly searching the existing possibilities. This contradiction in computer-aided inventing may be expressed as follows: the universe of possible variations should be widened so as to not be constrained to only the parameters of the object's original functional principle but should be constrained so as to reduce the search space to an affordable number of variants. Computer-aided inventing algorithms should be allowed to unrestrictedly search in the whole design space but as it rapidly grows to an infinite number of possible variations it should be constrained to such variations that allow a "maximal slope" of the conflicting objective parameters.

## 7.1 Contributions of the research work

In this work has been presented a procedural framework for a general strategy that extends the product development cycle into an innovation one. It takes the cases study for developing the geometry of a crankshaft using different tools as it is the genetic algorithms. The present research attempted to develop a set of guidelines to be integrated into a CAD platform in order to support the designer in the analysis of the conflicting features of the crankshaft under development and to provide systematic directions for the implementation of a solution that overcomes latent “geometrical contradictions”. In the pursuing of a new paradigm in product development, next are listed the main contributions of the research.

-The main contribution can be stated as a proposal of an integration loop inside the product development cycle, which merges the activities during concept generation and evaluation, together with detailed design in a way that allows the designer overcome mental inertia. The procedure results in a strategy that merges the integration loop into a general computer aided innovation framework, which compiles the elements that are part of the optimization tasks but are extended, in order to provide innovative results.

-The implementation of genetic algorithms in automatic optimization tasks involving the use of splines and the integration of CAD and CAE systems. CAD software to define the model geometry using splines, CAE to perform simulations of dynamic characteristics, process analysis and mathematical models of the product, controlled by a managing layer created in a programming code, used to integrate all these types of software.

-The generation of design concepts for the shape of the counterweights, which help the designer to develop a novel crankshaft. The concepts obtained were both different from what it has been found in the patents, and the proof of concept shows that the material flow during the forging process is able to fill the cavity of the die.

-The development of the initial approaches to find a correlation between the inventive principles form TRIZ, that complement the genetic operators (mutation, crossover, etc) of the genetic algorithms theory, in order to be able to generate a list of “extended operators” to apply to the geometry of products. The correlation is based on the natural convergence of the laws of technological evolution from TRIZ, and the laws of biological evolution, from where genetic algorithms take its origins.

For the attainment of the contributions, the research work consists of several chapters. An initial introductory chapter presented the concepts of evolutionary design and its use in optimization of mechanical systems; particularly the shape optimization based on genetic algorithms and the use of spline curves. The second chapter presented the cases study of a v-type 6-cylinder engine crankshaft; it is described an introduction to the dynamics of balance and a description of the balance equations. Third chapter describes the use of genetic algorithms in optimization and general characteristics of this type of algorithm; and introduces the general characteristics of the DAKOTA software that is used during the development of the case study.

In the fourth chapter there is a description of the conceptual framework for the creation of the strategy that extends the optimization paradigm into an innovation one. It is described how the standard product development cycle is enhanced. The steps of the strategy are described and the different optimization/innovation tasks presented. Chapter five contains the description of the deployment of every task implementation: from optimization of balance to a multi-objective optimization involving curvature and eigen-frequencies. The results of all tasks are presented, which leads to a solution of a crankshaft with some novel characteristics that are changing the paradigm in counterweight design, not described in previous patents.

In chapter six there are some thoughts on evolutionary design, genetic algorithms and inventive principles; at the same time, an algorithm to associate relevant inventive principles to models of geometrical contradictions has also being developed. Suggestions will guide the designer through the analysis of the conflicting geometrical features to the redefinition of the optimization tasks. There is an explanation of current optimization systems and of how different techniques, e.g. Pareto diagrams, can help identify conflicts. The aim was to resolve these conflicts by “innovation agents”. Further development of the optimization systems (parametric and topology) can only be achieved by integrating them with innovation methods. The resulting systems have a parametric shape and an innovative topological configuration. Some features need to be added to the system to be able to work in an "outside of the paradigm" way, and reach solutions that have not been considered before.

The use and combination of optimization tools and innovation capabilities is intended to provide a means for “automatically” varying the shapes, forms and topologies derived from the evaluation made by CAE systems. Genetic algorithms are viewed as mechanism for embedding the innovative principles in a CAD interface. For example the parametric optimization capabilities of CAD systems can be extended from simple “size variations” to an enhanced performance similar to what can be achieved by FEM but without losing the parametric functionality. With this insight into enabling creativity by evolution, a framework for explorative supporting creative design by evolutionary computing technology has emerged; for only a part of generated mathematical expressions can be expressed by curves and generate useful shapes. CAI is intended to stimulate the imagination of designers and activate their ‘mind’s eye’. It will give the designers concrete help to extend their design spaces

Performance enhancements are first achieved through optimization methods for finding maxima or minima of the response surface. Then, after performance enhancements through parametric changes are exhausted, paradigm shifts and other qualitative changes that lead to innovation are required. This is a step closer to building a Computer Aided Innovation Environment that goes more deeply into the evolution of technical systems as an analogy to biological evolution. The basics of a new concept of engineering design software and computer aided innovation (CAI), are presented and the impact on global design performance and limits exploration is analyzed. In particular, the ability to introduce automatic changes in shapes and topologies in a 3D-CAD environment is linked to the concept of CAI, allowing for an exploration of a broader field for possible solutions to a design problem. As a final reflection, it can be said that creativity and innovation can be structured in an objective methodology and taken out of the individual’s sub-conscience. Inventive principles suggest a series of recommendations to change the direction in which solutions are searched. These recommendations can be regarded as a knowledge database, which can be used to feed the cataclysmic similar transformation of genotypes during an evolution for optimization, allowing it to transcend the barriers of contradictions or constraints.

The use of GAs in this research allowed the design and optimization of crankshaft shape, but GAs also automated the design process. The design automation was one of the pursuits when the GA-based approach was used for crankshaft shape optimization, since the conventional counterweight shape determination processes are typically both laborious and time consuming. The evolutionary approach and the conventional trial-and-error method are both based on using the same simulator programs for solving the real-life design problems. The difference lies in their capabilities for optimization. However, it is neither relevant nor possible to conclude that the crankshaft designed by an evolutionary approach would perform better with respect to the corresponding physical reality. At any rate, it can be concluded that it is likely to perform better if the quality of the used simulation model is at least reasonably good.

Further research is necessary to implement the "automatic" variation of shape, topology and physical principles involved in a product development process following the recommendations derived from the simulation of the product performance parameters in a CAD/CAE environment. As the patterns of product evolution are useful in selecting the directions of possible variations to the functional principles further research is also required to look for the possible selection of alternative functional principles in a computer-aided inventing environment.

## 7.2 Future work

Two main directions of study are necessary to reach the attainment of an integrated approach for computer-aided innovation: The technical deployment and the theoretical foundation.

For the technical deployment, some future lines of evolution for computer aided innovation approaches include the integration of evolutive algorithms with functional models in order to simplify the transition from concepts to virtual prototypes. Functional analysis allows the designer to see the complete design intention in the early stages of the product development cycle. A good functional model should also provide integration with existing CAD models. It is necessary for instance to take into account the current CAD representation during functional modeling because it is one of the key elements to an automation process.

Methods to represent functions, either by statements or by systems, can be structured representations based on graphs. One benefit of the models on graphs is that there are many rules, properties, and algorithms for analysis derived from graph theory. The graph data structure is a robust tool for modeling engineering knowledge of geometry. In addition, the graphs provide the ability to model connections between different elements, and illustrate their relationships. This is essential for modeling behavior and functionality, needed to design a solution concept or knowledge gain.

Functional representation of models is not the only possibility to implement a technical approach. NIST Design Repository Project is a project of the National Institute of Standards and Technology (NIST) that involves research on providing a technical foundation for the creation of design repositories. Repositories of heterogeneous knowledge and data that are designated to support representation, capture, exchange, and reuse of design knowledge in general. One possibility for future work is to analyze the way in which design repositories can be implemented and used for the integration of the elements that comprise the CAI strategy, for example

- Development of a platform for modeling information to support the product engineering modeling; and provide a more comprehensive knowledge than traditional CAD systems.
- Implementation of interfaces for creating, editing, design and search design repositories that are easy to use and effective to convey the information you want, during a design process.
- The use of standard representations, when possible, and a contribution to the long-term standard developments where there is currently no evolutionary development

To achieve the technical implementation of CAI, we must work with existing CAD systems, especially those who make use of equations for modeling of products, added to the possibility of being integrated into a programming language. This is important since it will integrate the concepts of functional elements in a natural way. On the other hand, it is important to consider the basic geometry, because when changing the dimensions of the parameters, we could make use of genetic algorithms.

The possibility of an automatic development of the design process and product innovation, involving the above elements, requires a theoretical and a practical basis. CAD representations, model repositories, knowledge databases, spline curves, etc., must be integrated with analysis at a functional level, simulations, CAE, genetic algorithms and implementations particularly supported by high-level concepts like TRIZ. Given the level of computer processing, one can predict that the graphics and programming skills will perform the referred activities in the near future. The possibility of integrating mathematical methods such as genetic algorithms with CAD representations and a functional description that may relate to the principles of TRIZ, make it possible to obtain an automatic design and innovation process.

A future work is related to a profound analysis of the most effective genetic algorithms nowadays, and to implement the inventive principles in an algorithmic way so that the concepts of TRIZ be part of the genetic operators. There is still work to do in that direction but we must make the first steps.

A theoretical study is still necessary to pursue in the aim of the complete comprehension of the technical deployment. Questions like why using GAs instead of other methods? The answer may be linked to the versatility of evolutionary algorithms, in general, and GAs, in particular, makes them suitable for design-creativity tasks [2]. It has been expressed in chapter six how the evolution of technological systems and biological evolution are linked; it is reasonable to believe that the human evolution and the human creativity processes are also closely related. In situations where the borderline between optimization and creativity becomes undistinguishable, it is comprehensible to suppose that TRIZ and GAs are pointing in the same direction, but this is not enough to obtain a synergistic interaction. Some authors are proposing the use of dialectic laws as a framework to generate a coherent method integrating both tools as the basis for developing a computer-aided innovation framework [106].

Since evolution has proven to be a good general strategy for problem solving, the possible interaction with GAs and design methodologies is another future work that could be pursued. With GAs and TRIZ complementing each other, Evolutionary Innovation Axioms can be proposed:

- The phenotypic representation (observable characteristic) of the model can be hierarchically ordered, with the geometric dimension as its minimal level. The next level is its genotypic representation (chromosome: 0's and 1's).
- A conflict can be expressed as the win-loss relationship between two or more phenotypic characteristics of a product, denominated "target functions" of the system, linked by a unique genotype.
- The genetic operators act at a genotypic level during the evolution of a product, while the innovation operators act at a phenotypic level, which implies a *cataclysmic* change in the product characteristics.

In this context, the structure of inventive principles could be redefined, including analogies between GA operators and TRIZ. There are very few studies linking TRIZ and genetic algorithms and when approached, it is treated very superficially. Both methods, together, can be further studied in order to increase the performance of the innovation process, to optimize, predict and innovate. The dialectic approach could reorganize the inventive principles and extend them into a more axiomatic structure and make them compatible with the genetic algorithms operators. Future work is necessary to discover new inventive principles and classify them in levels of specification.

Since innovating requires seeing what others do not, the inclusion of new "search dimensions" will lead from optimization to innovation. From this viewpoint, the difference between traditional computerized optimization techniques and computer-aided innovation (CAI) could be done by considering the capability of expanding the search space for enhanced performance of the technological system, when the search process of the traditional optimization approach does not deliver the required performance enhancement. A computer-aided innovation framework could be defined as a new system which aids in identifying the most likely constraints of the optimization process. The cycle have to be surpassed or overcome as achieving the ideal performance; and, on the other hand , to be integrated into CAD/CAE systems that can expand autonomously the original search space into new areas, and look for "inventive solutions" likely to deliver the requested performance. Thus, a CAI framework is similar to two main ways human inventors look for new inventive solutions:

1. New conceptual approaches based on establishing new shapes, topologies or physical principles that add new capabilities to the existing systems.
2. Hybridizing the original system with others that add value to the new emerging one.

Some authors are attempting to develop a theory of Universal Scheme of Evolution [107]. According to the author, his theory in its structure is similar to the Genetic Algorithm approach. A Universal Scheme of Evolution is developed on the base of trends of Engineering Systems evolution. From this point, their similarity grows. As computer time becomes ever more inexpensive, researchers will start routinely use genetic programming to produce useful new designs, generate patentable new inventions, and engineer around existing patents. If assuming that computer intelligence is the ability of the program to reflect the trends of the surrounding world, and this knowledge is used to gain system's goals, then when the program actively interacts with its environment, it has the elements of intelligence, and acts in accordance with USE algorithm.

A more comprehensive approach is what it recently has been called C-K Theory. C-K Theory, or Concept-Knowledge Theory, is a theory of reasoning in design [108]. It defines design reasoning as a process of logic expansion, i.e. a logic that organizes the generation of unknown objects. C-K theory is also a design theory that explains cognitive and collective aspects of design activities by such specific mode of reasoning. According to the theory, the process of design is defined as a double expansion of the C and K (C a space of concepts, K a space of knowledge). That expansion is performed through the application of four types of operators: C→C, C→K, K→C, K→K. According to their authors, C-K theory has been presented with a limited mathematical development. Yet there is a large area of investigation in this direction. The properties of K can be studied in more detail and the structure of the four operators presents very interesting features. A future work can attempt to characterize the conditions that warrant the existence of disjunctions and conjunctions; and finally investigate the mathematical and computerized tools that could capture the C-K process.

One of the questions that arise is whether the man needs preparation to innovate. Or is it a computer that the one that must perform this task. The latter enters in the field of artificial intelligence, which has been one of the most controversial issues in the past 50 years. In between, we could make use of the computer to clarify our ideas and, by doing so, to achieve the goal of customer satisfaction and therefore our permanence in the market.

Experience and judgment can lead to good design. When evaluating a fitness function genetic algorithms rely only on the latter of these two characteristics (judgment), based on evaluation and comparison against certain criteria. The experience characteristic derived from the substantial knowledge of designers can be added to genetic algorithms by incorporating into the process inventive principles such as cataclysmic genetic operators. This additional information aids in the possibility of generating multiple functional variants that could be integrated into the creative process. Shape variations in CAD systems can contribute to stimulating the capacity of the designer's imagination to generate more design variants, thus reducing the psychological inertia [44].

# Chapter 8 Bibliography


[1] R. Dawkins, *The selfish gene*, Oxford, New York: Oxford University Press, 1989.

[2] P. Bentley, *Evolutionary Design by Computers*, USA: Morgan Kaufmann, 1999.

[3] R. Kicinger, T. Arciszewski, and K.D. Jong, "Evolutionary computation and structural design: A survey of the state-of-the-art," *Computers & Structures*, vol. 83, Sep. 2005, pp. 1943-1978.

[4] H. Kim, O.M. Querin, and G.P. Steven, "On the development of structural optimisation and its relevance in engineering design," *Design Studies*, vol. 23, Jan. 2002, pp. 85-102.

[5] Liu and Liu, "Creative Tower Generated by Computational Intelligence," *Trends in Computer Aided Innovation*, 2007.

[6] P.C. Matthews, L.T.M. Blessing, and K.M. Wallace, "The introduction of a design heuristics extraction method," *Advanced Engineering Informatics*, vol. 16, Jan. 2002, pp. 3-19.

[7] R. Oxman, "The thinking eye: visual re-cognition in design emergence," *Design Studies*, vol. 23, Mar. 2002, pp. 135-164.

[8] W.D. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," *Phys. D*, vol. 42, 1990, pp. 228-234.

[9] H. Liu, M. Tang, and J.H. Frazer, "Supporting creative design in a visual evolutionary computing environment," *Advances in Engineering Software*, vol. 35, May. 2004, pp. 261-271.

[10] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of MichiganPress, 1975.

[11] N. Leon-Rovira, D. Silva, J.M. Cueva de León, and J. Gutierrez, "Automatic Changes in Topology of Parts in 3D CAD systems," *1st IFIP TC5 Working Conference on Computer Aided Innovation*, Ulm, Germany.: 2005.

[12] K. Meniru, H. Rivard, and C. Bédard, "Specifications for computer-aided conceptual building design," *Design Studies*, vol. 24, Jan. 2003, pp. 51-71.

[13] Y. ZhouPing, D. Han, L. HanXiong, and X. YouLun, "A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies," *Computer-Aided Design*, vol. 35, Jan. 2003, pp. 37-56.

[14] G. Cascini, P. Rissone, and F. Rotini, "From design optimization systems to geometrical contradictions," *Proceedings of the TRIZ-Future Conference*, Germany: Kassel University Press, 2007, pp. 23-29.

[15] F. Capello and A. Mancuso, "A genetic algorithm for combined topology and shape optimisations," *Computer-Aided Design*, vol. 35, Jul. 2003, pp. 761-769.

[16] J. Lampinen, “Cam shape optimisation by genetic algorithm,” *Computer-Aided Design*, vol. 35, 2003, pp. 727-737.

[17] M. Haq and S. Rudolph, “A design language for generic space-frame structure design,” *Int. J. Comput. Appl. Technol.*, vol. 30, 2007, pp. 77-87.

[18] B.K. Belur and R.V. Grandhi, “Geometric deviations in forging and cooling operations due to process uncertainties,” *Journal of Materials Processing Technology*, vol. 152, Oct. 2004, pp. 204-214.

[19] C.A. Coello Coello and A.D. Christiansen, “An Approach to Multiobjective Optimization using Genetic Algorithms,” *Intelligent engineering systems through artificial neural networks.*, St. Louis, Missouri, U.S.A. : ASME Press, 1995, pp. 411-416.

[20] N. Leon-Rovira, J. Cueva, C. Villarreal, S. Hutron, and G. Campero, “Automatic shape variations for optimization and innovation,” *Trends in Computer Aided Innovation*, USA: (Boston: Springer), 2007, pp. 179-188.

[21] J.M. Widmann, “Structural shape optimization with intrinsic geometry and adaptive model refinement,” dissertation, Stanford University, 1994.

[22] N. Thiyagarajan and R.V. Grandhi, “Multi-level design process for 3-D preform shape optimization in metal forming,” *Journal of Materials Processing Technology*, vol. 170, Diciembre. 2005, pp. 421-429.

[23] M.H. Imam, “Three-dimensional shape optimization,” *International Journal for Numerical Methods in Engineering*, vol. 18, 1982, pp. 661-673.

[24] S. Kilian, U. Zandler, and F.E. Talke, “Suspension modeling and optimization using finite element analysis,” *Tribology International*, vol. 36, Jun. 2003, pp. 317-324.

[25] M.Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, Jan. 2003, pp. 227-246.

[26] S. Kota, J. Hetrick, Z. Li, and L. Saggere, “Tailoring unconventional actuators using compliant transmissions: design methods and applications,” *Mechatronics, IEEE/ASME Transactions on*, vol. 4, 1999, pp. 396-408.

[27] J.M. Cueva de León, “Automatic shape variations for optimization purposes,” master thesis, Tecnológico de Monterrey, Campus Monterrey, 2006.

[28] N. Leon-Rovira, J.M. Cueva de León, J. gutierrez, and D. silva, “Automatic Shape Variations in 3d CAD Environments,” *1st IFIP-TC5 Working Conference on Computer Aided Innovation*, Germany: 2005, pp. 200-210.

[29] R. Białecki, T. Burczyński, A. Długosz, W. Kuś, and Z. Ostrowski, “Evolutionary shape optimization of thermoelastic bodies exchanging heat by convection and radiation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, Apr. 2005, pp. 1839-1859.

[30] K. Suzuki and N. Kikuchi, “A homogenization method for shape and topology optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 93, Dec. 1991, pp. 291-318.

[31] L. Jackowska-Strumillo, J. Sokolowski, and A. Zochowski, “The topological derivative method in shape optimization,” *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, 1999, pp. 674-679 vol.1.

[32] C.A. Coello Coello, “A Comprehensive Survey of Evolutionary-Based Multi-objective Optimization Techniques,” *Knowledge and Information Systems*, vol. 1, 1999, pp. 129-156.

[33] B.E. Cohanim, J.N. Hewitt, and O. de Weck, “The Design of Radio Telescope Array Configurations using Multiobjective Optimization: Imaging Performance versus Cable Length,” *astro-ph/0405183*, 2004, pp. 1-42.

[34] S. Williams, “Unnatural Selection - Machines using genetic algorithms are better than humans at designing other machines,” *Technology review : MIT's magazine of innovation.*, vol. 108, 2005, p. 54.

[35] M. Olhofer, Yaochu Jin, and B. Sendhoff, “Adaptive encoding for aerodynamic shape optimization using evolution strategies,” *Evolutionary Computation*, Seoul: 2001, pp. 576-583.

[36] S. Obayashi, T. Tsukahara, and T. Nakamura, “Multiobjective genetic algorithm applied to aerodynamic design of cascade airfoils,” *Industrial Electronics, IEEE Transactions on*, vol. 47, 2000, pp. 211-216.

[37] S. Hyun, C. Kim, J.H. Son, S.H. Shin, and Y.S. Kim, “An efficient shape optimization method based on FEM and B-spline curves and shaping a torque converter clutch disk,” *Finite Elem. Anal. Des.*, vol. 40, 2004, pp. 1803-1815.

[38] C.C. Chang and A.N. Bramley, “Forging preform design using a reverse simulation approach with the upper bound finite element procedure,” *Proceedings of the I MECH E Part C Journal of Mechanical Engineering Science*, Professional Engineering Publishing, 2000, pp. 127-136(10).

[39] C. Chang and A. Bramley, “A New Forging Preform Design Approach Using Reverse Simulation,” *CIRP Annals - Manufacturing Technology*, vol. 47, 1998, pp. 193-196.

[40] T. Hughes, J. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, Oct. 2005, pp. 4135-4195.

[41] R.H. Bartels, J.C. Beatty, and B.A. Barsky, *An introduction to splines for use in computer graphics and geometric modeling*, 1995.

[42] N. Leon-Rovira, E. Uresti, and W. Arcos, “Fan-shape optimisation using CFD and genetic algorithms for increasing the efficiency of electric motors,” *Int. J. Comput. Appl. Technol.*, vol. 30, 2007, pp. 47-58.

[43] C. Kim and J. Yoon , “Design optimization of automotive lock-up clutches with damper ssprings using simulated annealing, fem, and b-spline curves,” *International journal of automotive technology*, vol. 8, Oct. 2007, pp. 599-603.

[44] N. Leon-Rovira, J.M. Cueva, D. Silva, and J. Gutierrez, “Automatic shape and topology variations in 3D CAD environments for genetic optimisation,” *Int. J. Comput. Appl. Technol.*, vol. 30, 2007, pp. 59-68.

[45] Fujiichi, Toshinobu, and Yoshihide, “Data fitting with a spline using a real-coded genetic algorithm,” *Computer-Aided Design*, vol. 35, Jul. 2003, pp. 751-760.

[46] K. Yang and B. El-Haik, *Design for Six Sigma a roadmap for product development*, New York: McGraw-Hill, 2003.

[47] Z.P. Mourelatos, “A crankshaft system model for structural dynamic analysis of internal combustion engines,” *Computers & Structures*, vol. 79, 2001, pp. 2009-2027.

[48] H.J. Henning, “Various processes for forging crankshafts,” *Forging*, Jul. 2003.

[49] H. Aguayo-Téllez and N. Leon-Rovira, “Computer Aided Innovation of Crankshafts Using Genetic Algorithms,” *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*, USA: Springer Boston, 2006, pp. 471-476.

[50] H. Aguayo-Téllez, N. Leon-Rovira, A. Albers, and T. Maier, “Comparison of Strategies for the Optimization/Innovation of Crankshaft Balance,” *Trends in Computer Aided Innovation,* USA: Springer, 2007, pp. 201-210.

[51] Y. Kang, M. Tseng, S. Wang, C. Chiang, and C. Wang, “An accuracy improvement for balancing crankshafts,” *Mechanism and Machine Theory,* vol. 38, 2003, pp. 1449-1467.

[52] C.M. Fonseca and P.J. Fleming, “An Overview of Evolutionary Algorithms in Multiobjective Optimization,” *Evolutionary Computation,* vol. 3, 1995, pp. 1-16.

[53] J. Dolza, G. Burrell, E.B. Sherrtck, and R. Schilling, “V-SIX ENGINE,” U.S. Patent 2632340, March 24, 1953.

[54] K. Hara and H. Shirose, “Crankshaft of V-6 internal combustion engine,” U.S. Patent 4552104, November 12, 1985.

[55] E. Ito and S. Fukazawa, “Engine crankshaft structure,” U.S. Patent 4730512, March 15, 1988.

[56] S. Baek, “Crankshaft of V-type 6-cylinder internal combustion engine,” U.S. Patent 5481942, January 9, 1996.

[57] H. Ozeki, “Crankshaft,” U.S. Patent 5758551, June 2, 1998.

[58] F. Ferrazzi, “Crankshaft for a v-type internal combustion engine,” U.S. Patent 7210373, May 1, 2007.

[59] T. Nagira, “Crankshaft for V-type six-cylinder engine,” U.S. Patent 7234432, June 26, 2007.

[60] T. Koike and T. Matsui, “Engine crankshaft made by forging,” U.S. Patent 6324942, December 4, 2001.

[61] S. Rao, *Mechanical vibrations,* USA: Addison-Wesley, 1990.

[62] C.A. Coello Coello, *An empirical study of evolutionary techniques for multi-objective optimization in engineering design,* USA: Tulane University, 1996.

[63] T. Weise, *Global Optimization Algorithms - Theory and Application,* Online as e-book: Thomas Weise, University of Kassel, Distributed Systems Group, 2008.

[64] T. Kovacs and P. Lanzi, “A Learning Classifier Systems Bibliography,” *Learning Classifier Systems,* 2000, pp. 321-347.

[65] J.H. Holland and A.W. Burks, “Method of controlling a classifier system,” U.S. Patent 4881178, November 14, 1989.

[66] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* USA: Addison-Wesley Longman Publishing Co., 1989.

[67] C. Darwin and H. Marcus, *The Origin of Species,* 1926.

[68] M. van der Steen and J. Enders, “Universities in Evolutionary Systems of Innovation,” *Creativity and Innovation Management,* vol. 17, Dec. 2008, pp. 281-292.

[69] V. Prushinskiy, G. Zainiev, and V. Gerasimov, *Hybridization,* Instantpublisher.Com, 2005.

[70] D.E. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” *Foundations of Genetic Alforithms,* 1991, pp. 69--93.

[71] J. Grefenstette, “Optimization of Control Parameters for Genetic Algorithms,” *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 16, 1986, pp. 122-128.

[72] K. Hayashi and M. Ashcraft, “Genetic design method and apparatus,” U.S. Patent 7542948, June 2, 2009.

[73] Valenzuela-Rendón, “The Virtual Gene Genetic Algorithm,” *Genetic and Evolutionary Computation – GECCO 2003,* 2003.

[74] D.A.V. Veldhuizen and G.B. Lamont, “Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art,” *Evolutionary Computation,* vol. 8, 2000, pp. 125-147.

[75] U. Cugini, G. Cascini, and M. Ugolotti, “Enhancing interoperability in the design process, the PROSIT approach,” *Trends in Computer Aided Innovation,* USA: (Boston: Springer), 2007, pp. 189-199.

[76] J.R. Koza, M.A. Keane, M.J. Streeter, T.P. Adams, and L.W. Jones, “Invention and creativity in automated design by means of genetic programming,” *Artif. Intell. Eng. Des. Anal. Manuf.,* vol. 18, 2004, pp. 245-269.

[77] C.A.C. Coello and A.D. Christiansen, “MOSES: A multiobjective optimization tool for engineering design ,” *Engineering Optimization,* vol. 31, 1999, p. 337.

[78] P.J. Bentley and J.P. Wakefield, *An Analysis of Multiobjective Optimization within Genetic Algorithms,* UK: 1996.

[79] N. Srinivas and K. Deb, “Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms,” *Evolutionary Computation,* vol. 2, 1994, pp. 221-248.

[80] J. Horn, N. Nafpliotis, and D. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization ,” *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on,* 1994, pp. 82-87 vol.1.

[81] A. Acan and A. Unveren, “Evolutionary multiobjective optimization with a segment based external memory support for the multiobjective quadratic assignment problem,”

*Evolutionary Computation, 2005. The 2005 IEEE Congress on,* 2005, pp. 2723-2729 Vol. 3.

[82] M. Eldred, A. Giunta, B.V. Bloemen, S. Wojtkiewicz, W. Hart, and M. Alleva, *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Reference Manual,* USA: Sandia Laboratories, 2002.

[83] K. Otto and K. Wood, *Product design : techniques in reverse engineering and new product development,* Upper Saddle River NJ: Prentice Hall, 2001.

[84] H. Aguayo-Téllez, "Modelo del proceso de diseño conceptual: integración de las metodologías QFD, análisis funcional y TRIZ," master thesis, ITESM, 1997.

[85] A.N. Bramley, "UBET and TEUBA: fast methods for forging simulation and preform design," *Journal of Materials Processing Technology, vol*. 116, Oct. 2001, pp. 62-66.

[86] C.H. Villarreal, "Optimization of the savonius wind turbine using a genetic algorithm," master thesis, Tecnológico de Monterrey, Campus Monterrey, 2008.

[87] R.E. Smith, B.A. Dike, and S.A. Stegmann, "Fitness inheritance in genetic algorithms," *ACM* s*ymposium on Applied computing, USA*: ACM, 1995, pp. 345-350.

[88] D.E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Spri*nger, 2002.

[89] N. Leon-Rovira, J. gutierrez, O. Martinez, and C. Castillo, "Optimization vs. Innovation in a CAE environment," *IFIP world computer congress 2004, Fr*ance: 2004.

[90] G. Altshuller, *Creativity as an exact science: the theory of the solution of inventive problems, New* York u.a.: Gordon and Breach, 1984.

[91] C.A. Coello Coello, "An updated survey of GA-based multi-objective optimization techniques," *ACM* Co*mputing Surveys, vol*. 32, 2000, pp. 109-143.

[92] G. Altshuller and L. Shulyak, *40 Principles: Triz Keys to Innovation, USA*: Technical Innovation Center, Inc, 1997.

[93] D. Cavallucci and François Rousselot, "Evolution Hypothesis as a means for linking system parameters and laws of engineering system evolution," *Proceedings of the TRIZ-Future Conference, Ger*many: Kassel University Press, 2007, pp. 31-40.

[94] Y. Zhang, M. Liao, and Z. Ren, "Cataclysmic Genetic Algorithm for Reactive Power Dispatch Comparing with an Interior-Point Algorithm," USA: M.H. Hamza, 2005.

[95] O. Dubrovsky, G. Levitin, and M. Penn, "A Genetic Algorithm with a Compact Solution Encoding for the Container Ship Stowage Problem," *Journal of Heuristics, vol*. 8, 2002, pp. 585-599.

[96] G. Levitin, J. Rubinovitz, and B. Shnits, "A genetic algorithm for robotic assembly line balancing," *European Journal of Operational Research, vol*. 168, Feb. 2006, pp. 811-825.

[97] H. Mizuyama and K. Ishida, "Systematic Decision Making Process for Identifying the Contradictions to be Tackled by TRIZ to Accomplish Product Innovation ," *Journal of Automation, Mobile Robotics & Intelligent Systems, vol*. 1, 2007, pp. 21-29.

[98] R. Dawkins, *The blind watchmaker, New* York: Norton, 1986.

[99] J. Gero, “Creativity, emergence and evolution in design,” vol. 9, 1992, pp. 435-448(14).

[100] M.D. Gross, “Emergence in a recognition based drawing interface,” *Proceedings of the 2nd international conference on visual and spatial reasoning, 2001*, pp. 51--65.

[101] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno, “Sub-part correspondence by structural descriptors of 3D shapes,” *Computer-Aided Design, vol*. 38, Sep. 2006, pp. 1002-1019.

[102] D. Bespalov, W.C. Regli, A. Shokouf, and eh, “Local feature extraction and matching partial objects,” *Computer-Aided Design, vol*. 38, Sep. 2006, pp. 1020-1037.

[103] H. Liu, “Generative Design Supported by Evolutionary Approach,” Ulm, Germany.: 2005.

[104] W.W. Hwa-Teh, “A knowledge based computational tool for creative conceptual design,” dissertation, Stanford University, 1992.

[105] M. Eisfeld and R. Scherer , “Assisting conceptual design of building structures by an interactive description logic based planner,” *Advanced Engineering Informatics, vol*. 17, Jan. 2003, pp. 41-57.

[106] R. Duran, N. Leon-Rovira, and E. Uresti, “A Dialectical Approach to Solve Inventive Problems Using Genetic Algorithms and TRIZ,” *Product and Systems Design, USA*: Design Society, 2009.

[107] A. Zakharov, “The Uniform Scheme of Evolution,” *The Second Annual AI TRIZ Conference Proceedings, USA*: AI TRIZ, , p. 183.

[108] A. Hatchuel and B. Weil, “A new approach to innovative design: an introduction to C-K theory,” Sweden: 2003.