Optimization with Genetic Algorithms and Splines as a way for Computer Aided Innovation

Follow up of an example with crankshafts

Albert Albers[1], Noel Leon Rovira[2], Humberto Aguayo[3], and Thomas Maier[4]

[1] Director of IPEK, Institute of Product Development, Universität Karlsruhe (TH), Germany, Email: albers@ipek.uni-karlsruhe.de

[2] Professor CIDT, Director Research Program Creativity and Innovation in Engineering, ITESM, Campus Monterrey, Chairperson WG5.4 Computer Aided Innovation IFIP. Mexico, Email: noel.leon@itesm.mx Phone: +52 81 81582012.

[3] Center for Innovation in Design & Technology (CIDT), ITESM, Mexico, Email: haguayo@itesm.mx

[4] IPEK, Institute of Product Development, Universität Karlsruhe (TH), Germany, Email: maier@ipek.uni-karlsruhe.de

Abstract: This paper describes the conceptual foundations to construct a method on Computer Aided Innovation for product development. It begins with a brief recap of the different methodologies and disciplines that build its bases. Evolutionary Design is presented and explained how the first activities in Genetic Algorithms (GAs) helped to produce computer shapes that resembled a creative behavior. A description of optimization processes based on Genetic Algorithms is presented, and some of the genetic operators are explained as a background of the creative operators that are intended to be developed. A summary of some Design Optimization Systems is also explained and its use of splined profiles to optimize mechanical structures. The approach to multi-objective optimization with Genetic Algorithms is analyzed from the point of view of Pareto diagrams. It is discussed how the transition from a multi-objective optimization conflict to a solution with the aim of an ideal result can be developed means the help of TRIZ (Theory of Inventive Problem Solving), complementing the discipline of Evolutionary Design. Similarities between Genetic Algorithms and TRIZ regarding ideality and evolution are identified and presented. Finally, a brief presentation of a case study about the design of engine crankshafts is used to explain the concepts and methods deployed. The authors have been working on strategies to optimize the balance of a crankshaft using CAD and CAE software, splines, Genetic Algorithms, and tools for its integration [1] [2].

Keywords: Genetic Algorithms, Splines, imbalance, TRIZ

1. Introduction

Computer Aided Innovation builds its bases on software tools used for a large number of applications: from modeling activities and optimization tasks, to performance's simulation of a product. But the addition of new tools is intended to extend the support to the creative part of the design process. This support allows the designer to improve the performance of their concepts, allowing computers to take part on the generation of variants, and on the judgment, by simulation, of these variants. Genetic Algorithms, CAD/CAE, Splines and TRIZ are all software tools that can nurture the knowledge of designers to generate new solutions, based on many separate ideas, suggesting entirely new design concepts. Methods for structural and topological optimization, based on evolutionary design, are used to obtain optimal geometric solutions. They are evolving to configurations that minimize the cost of trial and error and perform far beyond the abilities of the most skilled designer. Next is presented a brief description of the methods and tools that lead to our strategy of Computer Aided Innovation.

1.1. Evolutionary Design

A relatively new area of development called Evolutionary Design [3] is being object of intensive research. Peter Bentley describes that Evolutionary Design has its roots in computer science, design and evolutionary biology. It is a branch of evolutionary computation that 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. A huge variety of different engineering designs have been successfully optimized, using these methods. 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 parametrize those parts of the design they feel need improvement. Different brands of Evolutionary Design derive: Evolutionary Optimization, Creative Evolutionary Design and Conceptual Evolutionary Design. Evolutionary Optimization places great emphasis upon finding a solution as close to the global optimal as possible perhaps more than any other type of evolutionary design. Creative Evolutionary Design is concerned with the preliminary stages of the design process. But generating creative designs could only be possible by going beyond the bounds of a representation, and by finding a novel solution which simply could not have been defined by that representation. In Conceptual Evolutionary Design, the relationships and arrangements of high-level design concepts are evolved in an attempt to generate novel preliminary designs. Generative (or conceptual) Evolutionary Designs using computers to generate the form of designs rather than a collection of predefined high-level concepts has the advantage of giving greater freedom to the com-

puter. 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. Genetic Algorithms, an evolutionary computational tool, is selected to be integrated as part of our strategy.

1.2. Genetic Algorithms

Genetic Algorithms are global optimization techniques that avoid many of the shortcomings exhibited by local search techniques on difficult search spaces [4]. A GA is an iterative procedure which maintains a constant-size population P(t) of candidate solutions. During each iteration step, called a generation, the structures in the current population are evaluated, and, on the basis 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. In order to search other points in the search space, some variation is introduced into the new population by means of idealized genetic recombination operators. The most important recombination operator is called crossover. Under the crossover operator, two structures in the new population exchange portions of their internal representation. The power of GA's derives largely from their ability to exploit efficiently this vast amount of accumulating knowledge by means of 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 GA's exhibit impressive efficiency in practice. While classical gradient search techniques are more efficient for problems which satisfy tight constraints, GA's 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.

The class of GA's is distinguished from other optimization techniques by the use of concepts from population genetics to guide the search. However, like other classes of algorithms, GA's differ from one another with respect to several parameters and strategies:

1) Population Size (N): The population size affects both the ultimate performance and the efficiency of GA's. GA's generally do poorly with very small populations, because the population provides an insufficient sample size for most representations.

2) Crossover Rate (C): The crossover rate controls the frequency with which the crossover operator is applied. In each new population, C * N structures un-

dergo crossover. The higher the crossover rate, the more quickly new structures are introduced into the population.

3) 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.

4) 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).

5) Scaling Window (W): When maximizing a numerical function f(x) with a GA, it is common to define the performance value u(x) of a structure x as u(x) = f(x) - fmin, where fmin is the minimum value that f(x) can assume in the given search space.

6) 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. The optimization systems of our interest are described in the next section.

1.3. Design Optimization Systems

The evolution of Product Development tools has been characterized by different trends; the analysis of these trends offers useful hints for the prediction of next generation systems. In mechanical design, optimization tasks are used for structural optimization, which deals with the development of mechanical structures. For example, when minimizing the weight of the wing of an airplane or optimizing the shape of a crankshaft, restrictions have to be included to guarantee the stability of the structure (ex. stresses or natural frequencies). The objectives of structural optimization are: minimizing stress or weight; maximizing lifespan, stiffness or first natural frequency. Any of those under different constrains 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, position of nodes, shell thickness, beam cross-section, angle of fibers from compound materials, etc. As design variable restrictions we 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 possible. Particularly, two kinds of structural optimization are frequently used: Topology Optimization and Shape Optimization.

Topology Optimization consists on determining an optimal material distribution of a mechanical product. A basic FE model is created and analyzed in a design area with given boundary conditions. The aims are commonly to maximize stiffness or maximize the natural frequency of a product. The constraints of the

design are: the fixations, material volume and maximum displacement allowed. The design variables are the material density of the elements, which are counted commonly in hundreds of thousands; this means a huge amount of design variables. The goal is, given a predefined design domain in the 2D/3D space with structural boundary conditions and load definitions, to distribute a given mass, which is a given percentage of the initial mass in the domain such that a global measure takes a minimum (maximum) value.

Shape Optimization consists of changing the external borders of a mechanical component. The aims are: minimizing the stress or the volume or maximizing the natural frequency. Constrains to the design are: fixations, restrictions for displacement of component borders. The design variables of the product are, for geometric models: length, angle and radii measurements; for FE model: node co-ordinates.

Each optimization method uses a strategy to obtain 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 are: the number of variables (often a huge number of them); characteristics of the objective function (continuous, discontinuous, linear/quadratic/arbitrary, etc.); characteristics of restrictions (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.

In order to further develop the optimization systems it is required to add new concepts into the previous paradigms. A new kind of parameterization is inferred by taking the characteristics of last optimization methods. In order to obtain a similar behavior 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.

1.4. Design optimization of splines shapes

A great variety of different engineering designs have been successfully optimized using Evolutionary Design, i.e. antennas and aircraft geometries. Although the methods used by developers of such systems varies, one of these types of evolutionary design that has potential to be classified as generative or creative is the splined shape approach [5]. The splining of the shapes and its control points, codified to be interpreted by Genetic Algorithms, are the basis for an evolutionary designed shape. Practitioners of evolutionary optimization using splines usually start the process with an existing design, and then parameterize the control points of the splines that embody those parts of the design they feel need improvement. Moreover, the concept can be extended to reach the whole structure of the product and

even the functional structure. The control points are encoded as genes, the alleles (values) from which parameters are described, are evolved by an evolutionary search algorithm, i.e. Genetic Algorithms. Three main genetic operators act on the "genes" of the geometry, as known: selection, crossover and mutation. Crossover allows the geometrical characteristics of selected splines (compared from a fitness function) be merged in pairs and extend its properties to next generations. The designs are often judged by making an interface of the system to simulation or analysis software, which is used to obtain a fitness measure for each design.

2. Evolutionary Design transition to Computer Aided Innovation

In the previous section a brief explanation of the methods and tools that conducted our research work to the development of our framework on Computer Aided Innovation was presented. Starting from the Evolutionary Design approach, and particularly on Genetic Algorithms, the concept of splining applied to the structural optimization of products was explained. The last element to be considered is the analysis of conflicts during optimization that prevent a design to reach the Ideal Solution

2.1. Multi-objective Optimization and conflicts in product development

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. But 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. Typically, the GA user finds some ad-hoc function of the multiple attributes to yield a scalar fitness function. Often-seen tools for combining multiple attributes are constraints, with associated thresholds and penalty functions, and weights for linear combinations of attribute values. A few studies have tried a different approach to multi-criteria optimization with GAS: using the GAs to find all possible trade-offs among the multiple, conflicting objectives. Some authors propose to perform a set of mono-objective optimization tasks to reveal conflicts [6]. 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 lie on a surface known as the Pareto optimal frontier. The goal of a Pareto is to find and maintain a representative sampling of solutions on the Pareto front. Hence, the term "opti-

mize" is referred to find a solution, which would give the values of all the objective functions an "acceptable trade off" to the designer [7]. Moreover, computer geneticists have faced the concept of the ideal [8]. They named it the ideal point. The Pareto diagram (used mainly in multi-objective optimization processes) shows a boundary that divides the region of feasible solutions from the point where the ideal solution lies. When there is a set of optimal solutions lying on a line that prevent the functions to reach the "ideal" at the same time, because of constraints in the solution space, it becomes an unrealistic goal to reach the ideal point.
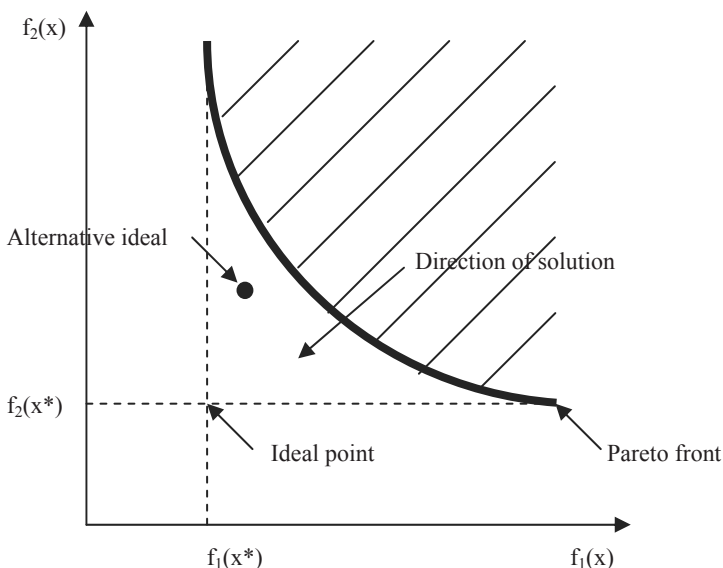

Figure 1. Pareto diagram and the concept of ideal

According to traditional TRIZ theory, the reach of an Ideal Final Result is encouraged and TRIZ presents tools for identifying technical and physical contradictions underlying 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" [9]. It is a natural convergence direction to merge Evolutionary Design (based on laws of biological evolution) with TRIZ (based on laws of technical evolution) inside a computer framework aimed to Computer Aided Innovation.

GAs can extend its paradigm of multi-objective optimization by taking advantage of the inventive principles, letting the operators be not only the basic "muta-

tion" and "crossover" but new operators or "agents" capable to modify the way the algorithms perform on the CAD geometry [10]. In this way, CAD systems could develop new configurations and alternative modifications to the geometry, in order to reach the ideal point or the "Ideal Final Result". The solution can reach a level of detail that derives in the possibility that the designer be inspired by these suggestions, selects the most suitable solution and implements it. In other words the designer could be presented a set of alternative modifications, defined automatically on the base of the selected principles that may be applied based on the concept of "Cataclysm Mutations" [11]. Cataclysmic mutations with similar pattern are now being studied in Evolutionary Algorithms as tools for finding innovations [12][13]. From the TRIZ inventive principles, those that have a geometric interpretation can be added to form the extended cataclysmic operators. See table 1.

Table 1. Genetic interpretation of TRIZ inventive principles

TRIZ principles	Genetic Interpretation
Segmentation, combination	Divide two genotypes and combine alternate parts (Crossover, simple)
Asymmetry	Break symmetrical genotypes (Crossover, simple)
Merging	Join genotypes (Crossover, simple)
Nesting	Place part of a genotype inside another (Crossover, nesting)
Another dimension	Create genotypes from different parameters (Crossover, nesting)
Homogeneity	Turn a genotype homogeneous (Crossover, nesting)
Discarding and recovering	Break and rebuild genotypes (Crossover, nesting)
Inversion	Turn around a genotype (Inversion, genetic)
Extraction	Extract a gen in a genotype (Mutation)
Feedback	Return fittest genotypes (Selection)
Copying	Take a copy of fittest genotypes (Selection)

The level of impact from the different operators can vary from a slow and steady accumulation of changes (the way an optimization algorithm normally performs), to a sudden disturbance in the nature of the system (or cataclysm). The most important effect is creating a jump in the phase transition. More suggestions can be enriched by means of guidelines, provided by the inventive principles that can be associated to the genetic operators. As result, the algorithm should be capable of applying the agents according to the conflict that is being faced.

2.2. Follow up of crankshaft example

In an attempt to exemplify the concepts deployed, the development of an engine crankshaft is conducted by making automatic changes in the geometry of its counterweights. In order to make geometry modifications to our case study, the geometry of the counterweights was transformed from simple lines and arcs to spline curves. Splines allow smooth shape changes via the coordinates of its control points. That smooth shapes benefit the material fluency during the manufacturing process. The variation of these control points results in a balance response of the crankshaft. The x and y coordinates of the control points can be parametrically manipulated by the Genetic Algorithm. Figure 2 shows how the splines substitute the original profile of the crankshaft. It is possible to see how close the spline is to the original profile.

Counterweight 1   Counterweight 8
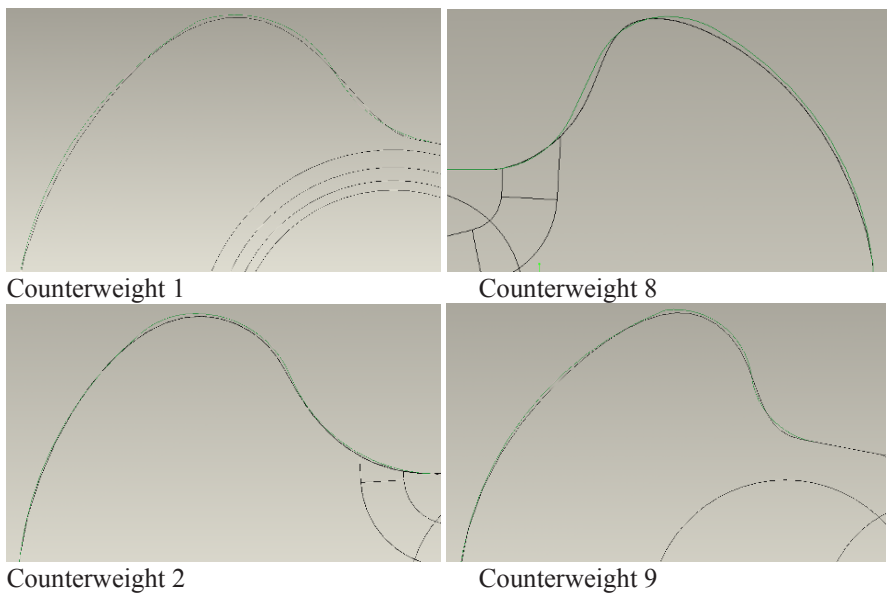
Counterweight 2   Counterweight 9

Figure 2. Splinization approach applied to a Crankshaft; the original profile consists of arc and line segments.

The selected Genetic Algorithms that were applied are from the DAKOTA toolkit from Sandia Laboratories. It was developed an interface programmed in Java language to link the GAs to the CAD geometry. The optimization loop runs fully automated so the computer generates shapes in every generation. Some of the genetic operators are described next.

It was used an initial population size of 50 individuals, because the number of genes were 32 (8 in each of the 4 counterweights) and it allows a good representation of the chromosomes in every generation. A crossover rate of 0.8 was selected, that was a number of 40 individuals out of 50 to be crossed over and have a wide amount of new shapes. A mutation rate of 0.1 allowed exploring the solution space for local optima not possible to find by conventional methods.

The results from the first attempts show that the imbalances from both sides of the crankshaft are in conflict each other. These conflicts are then aimed to be resolved by "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 in order to work in an "out of the paradigm" way, leading to solutions that were not considered before. In order to have a visual impression about the way the algorithm is performing, some of the counterweights are presented in the figure.
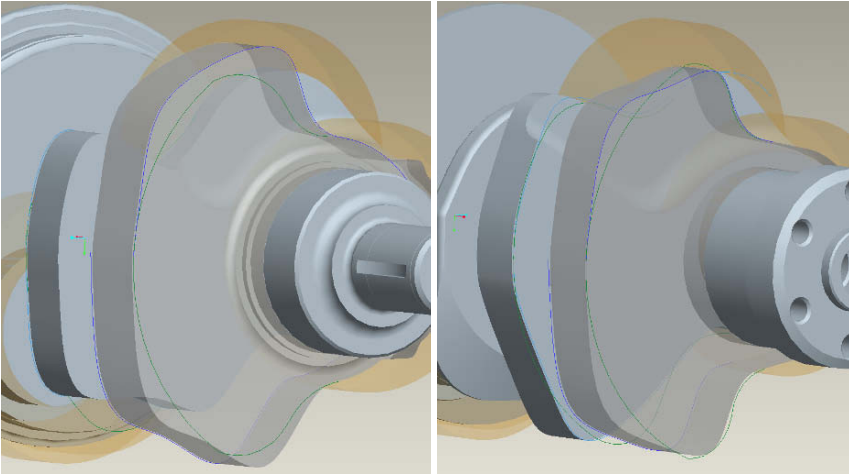

Figure 3. Representations of the crankshaft's counterweights (external ones, 1 and 9, are transparent to let visualize the others, 2 and 8)

The shapes are presenting some notches that are not suitable for forging, but the direction of 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 paper, presenting to the designer challenging alternatives. Finally, these proposals are solution triggers that inspire him, but they are not substituting its role in selecting the most suitable solutions and implement them properly.

3. Conclusions

This paper started with a brief recap of the different methodologies and disciplines that build the bases for developing a conceptual framework on Computer Aided Innovation. The area of Evolutionary Design is presented and explained how the first activities in Genetic Algorithms helped to produce the first computer shapes that resembled a creative behavior. Some of the genetic operators are explained as a background of the creative operators that are intended to be developed. A summary of some Design Optimization Systems is explained as also its use of spline profiles to optimize mechanical structures. The transition from a multi-objective optimization conflict to a solution with the aim of an ideal result is developed means the help of TRIZ. The innovative operators are analyzed to find relation with the genetic operators and turn into a "cataclysmic similar" set of new principles. Finally, an example of the development of an engine crankshaft is shown, with some preliminary results that may help to embody the complete framework of Computer Aided Innovation. Activities to be continued in the future are the definition of additional fitness functions not only in CAD but in CAE simulation (forging simulation), in order to control "strange" shapes. Also, objective functions and restrictions are going to be added by the use of forging simulation and stress analysis during geometry variations, resulting on what is pretended to be an integration of different systems running totally or partially automatic.

As a final reflection, it can be said that creativity and innovation can be structured to an objective methodology, and taken away from the individual's subconscience. 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 trespass the barriers of contradictions or constraints.

Experience and judgment can make a good design. When evaluating a fitness function, the genetic algorithms rely only in the last of these two characteristics (judgment) based on evaluation and comparison against certain criteria. The first one (experience) can be added from the substantial knowledge of designers into the genetic algorithms by means of the incorporation of inventive principles as cataclysm genetic operators.

4. Acknowledgments

References

1. Aguayo-Téllez H. et al.: Comparison of Strategies for the Optimization/Innovation of Crankshaft Balance. In Trends in Computer Aided Innovation, pp. 201-210, Springer, USA (2007).

2. Aguayo-Téllez H. and Leon-Rovira N.: Computer Aided Innovation of Crankshafts Using Genetic Algorithms, in Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management, pp. 471-476, USA: Springer Boston, (2006); http://dx.doi.org/10.1007/0-387-34403-9_64.

3. Bentley P.: Evolutionary Design by Computers, Morgan Kaufmann, (1999).

4. Goldberg D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co. USA Inc. (1989).

5. Leon-Rovira et al.: Automatic shape variations for optimization and innovation, in Trends in Computer Aided Innovation, pp. 179-188, USA, (2007); http://dx.doi.org/10.1007/978-0-387-75456-7_18.

6. Cugini, Cascini, Ugolotti: Enhancing interoperability in the design process, the PROSIT approach, in Trends in Computer Aided Innovation. USA, (2007); http://dx.doi.org/10.1007/978-0-387-75456-7_19.

7. Coello C.A.: An empirical study of evolutionary techniques for multi-objective optimization in engineering design, USA, Tulane University, (1996).

8. Coello C.A.: An updated survey of GA-based multi-objective optimization techniques. ACM Computing Surveys, vol. 32, pp. 109-143 (2000).

9. Altshuller G., Shulyak L.: 40 Principles: Triz Keys to Innovation. Technical Innovation Center, USA, Inc (1997).

10. Leon-Rovira N. et al.: Optimization vs. Innovation in a CAE environment, in IFIP world computer congress 2004, France: (2004).

11. Zhang Y., Liao M., and Ren Z.: Cataclysmic Genetic Algorithm for Reactive Power Dispatch Comparing with an Interior-Point Algorithm, M.H. Hamza, USA, (2005); http://www.actapress.com/Abstract.aspx?paperId=22063.

12. Dubrovsky O., Levitin G., Penn M.: A Genetic Algorithm with a Compact Solution Encoding for the Container Ship Stowage Problem, Journal of Heuristics, vol. 8, pp. 585-599, (2002).

13. Levitin G., Rubinovitz J., and Shnits B.: A genetic algorithm for robotic assembly line balancing, European Journal of Operational Research, vol. 168, pp. 811-825, (2006).