Toward natural optimization into CAD software

Or how to simply integrate an optimization tool into a CAD software for solving a whole class of problems

Jean-Christophe FAUROUX
LaRAMA / IFMA
Campus de Clermont-Ferrand / Les Cézeaux, BP 265, 63175 AUBIERE Cedex, France
E-Mail : fauroux@ifma.fr Tel : (33.) 4.73.28.80.50
Fax : (33.) 4.73.28.81.00

Abstract : In this paper, a new way of integrating optimization and CAD tools is presented. It permits a preliminary design of whole classes of mechanisms that would be too tedious to solve by other methods. Through the example of geared speed reducer design, numerous advantages can be pointed out : strong decrease in programming time, generic expression of the problem and improvement of interactivity.

Résumé : Ce travail propose une nouvelle façon d'intégrer des outils d'optimisation et de CAO afin de permettre la pré-conception optimale de classes entières de mécanismes qu'il serait trop fastidieux de traiter autrement. A travers l'exemple de la conception de réducteurs de vitesse à engrenages, nous montrons les nombreux avantages d'une telle démarche : réduction du temps d'écriture du programme, expression générique du problème et amélioration de l'interactivité.

1. Introduction

During the nineties, Computer Aided Design tools have been becoming very popular and common within engineering and design departments. They considerably facilitate the draftsman work and some of them even offer powerful calculation functions using the Finite Element Method (FEM). However, there is still a lack of CAD tools giving the opportunity to proceed to optimization calculations. This is a bit surprising as optimizing a product before its launching is a major concern for most of manufacturers. As it is pointed out in [1], it seems that optimization packages should be used much widely in the future, with more user-friendly graphical interfaces and a deeper integration into CAD software.

Among the few existing products that offer optimization capabilities for design, some of them propose to optimize a structure using FEM. This roughly means that a FEM calculation is performed at each iteration of the optimization process in order to optimize the static or dynamic behavior of the studied system. This seems to become a strong trend in the market [2][3] but is not our concern, even if it is a seducing way. We think that opportunities remain for developing a preliminary design tool that would not use intensive FEM calculations on a highly detailed structure but would rather help the user upstream in the design process, when designers are still using unrefined sketches.

In the following paper, we present in detail our vision of the way to integrate optimization and CAD tools in such a preliminary design software. An application example is given [4] with specific information on how we managed for the best coupling before both tools.

2. Nature of the problem

Optimization tools are generally considered to be difficult to use by non-specialists and to request for a rigorous formalism. For constructing a mechanical optimization problem, the classical following process has to be performed [5][6] :

1. Finding a model of the mechanical problem
2. Typing the characteristic relations of the model with the specific formalism requested by the optimization tool (objective function to be minimized, constraints expressed in a non dimensional way, numerical method to be used, parameters of this method...)
3. Compiling these data and linking them to the optimization library
4. Running the resulting optimization program
5. Reading results
6. Graphically representing the corresponding mechanical model using the newfound optimized numerical values

On one hand, this process is quite suited for solving a specific problem. For example, it was extensively applied for treating the typical problem of optimizing the weight or overall dimensions of a geared speed reducer [7][8][9][10]. The mechanism to be optimized is generally rather simple and has always a standard structure like parallel shafts and one or more spur gears.

On the other hand, it demands a great amount of work for solving more general problems. In our case, the process does not fit our work at all [4][11][12][13]. The purpose of this work is also to optimize geared speed reducers, but not necessarily of the same structure. Every type of speed reducer, made of a certain number of serially connected stages, can be considered. Of course, the stages can be of different nature : cylindrical gear (with external or internal contact), bevel gear, worm gear, warped gear... This leads to a more general optimization problem where architecture varies, components do not always have the same location, are not of the same type all the times and therefore obey various technological relations.

Consequently, it is not an isolated problem but rather a complete class of optimization problems that can be constructed. These problems are similar according to several points :

- The *objective function* is always the same. A criterion like overall dimensions or cost has to be minimized.
- They always involve numerous constraints :
  - Some of them are ***geometrical*** (part *non-interference, continuity* between stages, *closure condition* of the mechanism, ...)
  - Other ones are of ***technological*** nature (satisfying a given *speed ratio* or *efficiency*, ensuring part or assembly *resistance*, ...)
  - Some act at a ***local level*** : some of the constraints can be calculated knowing only local characteristics of one stage. In this category can be found constraints like tooth resistance or maximum tangential speed.
  - Other ones act at a ***global level*** : the closure, speed ratio or efficiency constraints are global ones. They are closely depending on nature and dimensions of each of the stages of the mechanism.

Thus, the constructing process of the optimization problem may become ***extremely tedious***, particularly at step 2. Actually, optimizing a new speed reducer induces numerous changes in the problem, as the number and shape of parts are modified. So *geometrical* constraints have to be entirely reconstructed. *Global technological* constraints must also be rewritten : some parameters like the speed ratio change when an external cylindrical gear has to be replaced by an internal cylindrical one. Only a few *local technological* constraints remain unchanged : for instance, calculation rules for a cylindrical gear do not change.

Step 6 may also be quite boring because it first requires to create a CAD model of the mechanism, then to regenerate it manually after each optimization for updating the newly calculated dimensions. Modern CAD tools offer convenient functions for parametered design, which should facilitate the regeneration operation. However, the initial building of a correct CAD model remains a delicate operation.

3. Solving method

The following section deals with the practical implementation of the method which was used for solving the general speed reducer preliminary design problem. The work was based on our self-developed experimental design software named CASYMIR ("**C**onception **A**ssistée de **SY**stèmes **M**écaniques de transm**I**ssion en **R**otation", which stands for "Assisted Design of Rotation Transmission Mechanical Systems) [13]. Our goal was to demonstrate the validity of the concept of integration between optimization and CAD tools and to show that it may considerably ease the work of the user at the early beginning of the design process. Therefore, a proprietary CAD tool was preferred to a commercial one for its better portability, stability in time and internal documentation.

For illustration purpose, the example of an industrial speed reducer design, already treated in [4], will be used. For this very concrete technological problem, a *convenient and generic formulation* will be presented for solving this problem as well as the whole associated class of problems. In four points, here are the main advantages of our method.

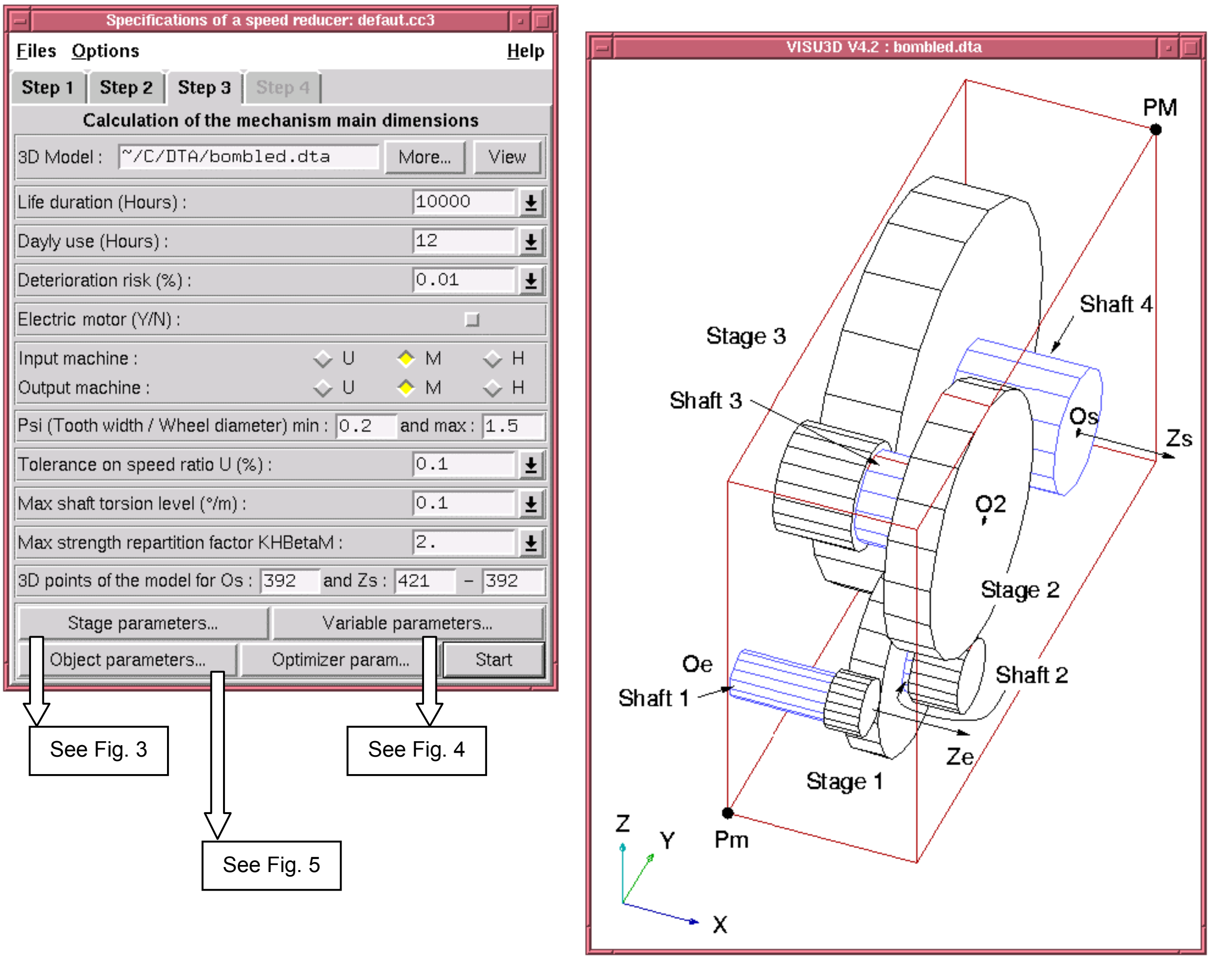

Fig. 1 : Example of the speed reducer to be optimized (On the left : specifications – On the right : CAD model)

3.1. Automatic mechanism drawing

First of all, the geometry of the mechanism to be optimized must be represented and modeled. Here can be seen a 3D model of the industrial speed reducer to be engineered which was constructed with our proprietary CAD tool named VISU3D (Fig. 1). Any other commercial CAD software might be suitable for this use.

An *object oriented* model was used for internal representation : the speed reducer is made of several stages, each of them being made of several parts (shafts, gears) that all have dimensions, 3D positions, 3D orientations and other attributes. Moreover, this is a fully *parametered model*, that is to say every part radius or shaft length may be modified, as well as the relative angular positions between consecutive stages. Finally, the model ensures the *mechanism continuity* : for instance, every dimension change in stage 1 has repercussions on the position of stages 2 and 3.

The model is automatically generated by CASYMIR thanks to a preliminary study of the best mechanism topologies answering given specifications [12]. This means that two tasks are automatically performed :

- Choice of the number, order and nature of the involved stages. It should be noted that the given example is made of only cylindrical gear stages but many other mechanisms are available (Fig. 2).
- The assembly of the geometrical models of stages as well as the grouping of internal parameters are fully treated by CASYMIR so the user does not have to work manually anymore.

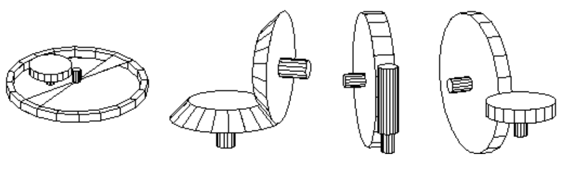

Fig. 2 : Four different types of stages (internal cylindrical, bevel, worm and warped gear)

3.2. Linking CAD and optimization tools

Above all, the considered problem is of geometrical nature. The expected solution is the vector of problem variables (dimensions and angles) minimizing overall dimensions. For every given vector of variables exists a corresponding instance of the CAD model. It follows that the model is a good and sufficient representation of the current state of the optimization problem.

As a natural consequence came the idea below : why not exploit the respective advantages of CAD and optimization software ? Each of them should deal with a specific task :

- The *CAD software* will store the current state of the mechanism model. It will perform the graphical representation and updating of the model when dimensions vary.
- The *optimization software* has just to read the current variable vector in the CAD model, to use a numerical method for improving the variable vector at the next iteration and finally re-inject the values in the CAD model.

Such a coupling offers numerous advantages :

- It avoids to store data in duplicate (mechanism equations AND also CAD model).
- User has no more to write complex geometrical relations : in order to obtain the wheel position of stage 2, the optimizing tool has just to read coordinates of point O2 (Fig. 1) in the CAD tool memory, which has already performed the tedious but indispensable numerous coordinate changes.
- The constraint writing step becomes more *simplified and generic* : when nature of the mechanism changes, no need for the user to rewrite constraints because they are automatically generated (cf. next subsection).

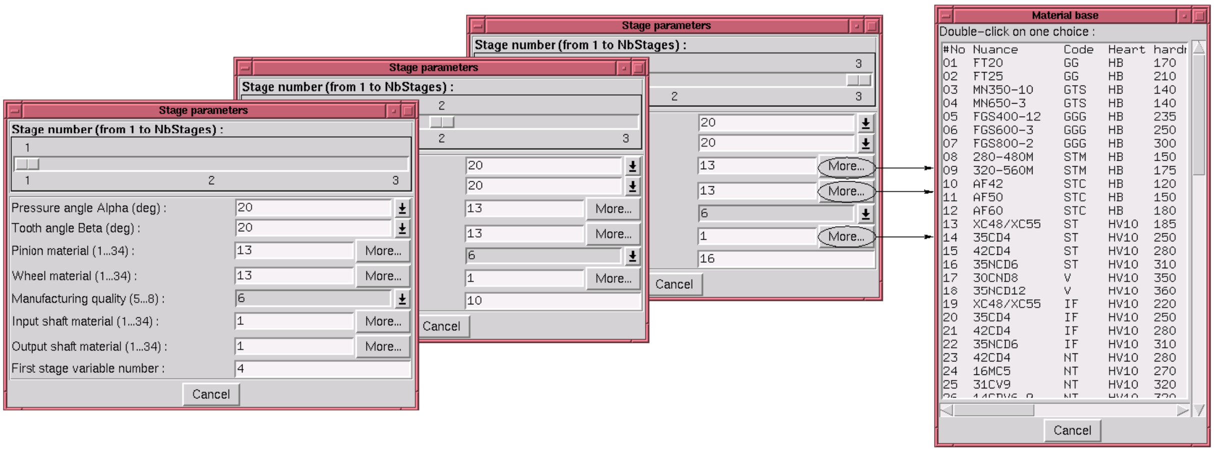

Fig. 3 : Property windows of each of the 3 speed reducer stages with the associated material database

From a concrete point of view, user must type the following data within various windows of the CASYMIR software :

- Characteristics of mechanism stages : gear tooth parameters, material, manufacturing quality (Fig. 3).
- Nature of variables, associated description and above all, address of the variable within the CAD software memory for good communication with the optimization tool (Fig. 4).

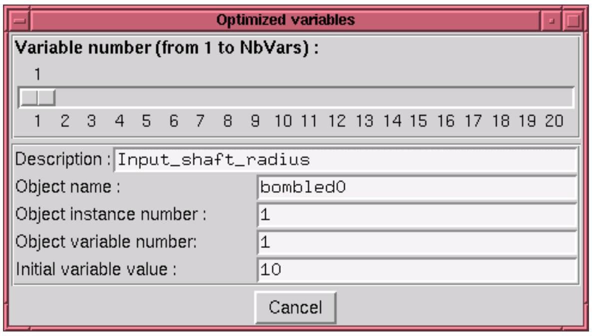

Fig. 4 : Properties of variable 1
(Input shaft radius)

- Nature of each of the objects the model is made of (cone, truncated cone, crown wheel), associated characteristic points, dimensional variables (Fig. 5).

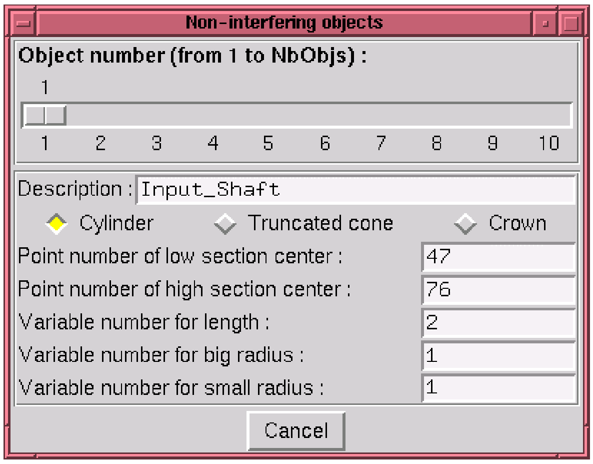

Fig. 5 :Properties of object 1 (Input shaft)

3.3. Automatic constraint construction

Technological constraints, depending on the nature of each stage, are stored in a constraint database in C function form. The functions may be modified independently. They are also added or removed each time a basic type of stage is added or removed in the mechanism database. CASYMIR is already provided with a default set of elementary stages and their associated constraints. The users has nothing to do except in the case where a totally new type of stage should be considered. Each constraint function gets input values (stage dimensions, input and output speeds of the stage, etc.) and returns results (constraint values, contribution of the stage to global constraints, etc.) as represented in Fig. 6.

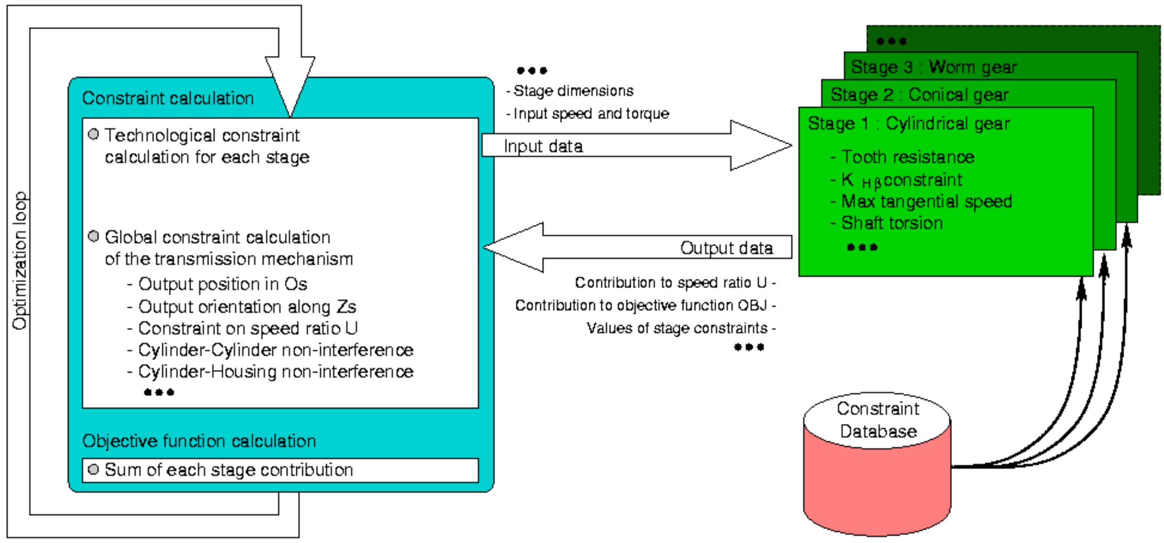

Fig. 6 : Automatic construction of constraints

User is liberated from writing any other constraint :

- *Global constraints*, concerning the whole mechanism. For instance, the global speed ratio *U* is obtained by multiplying all the intermediate stage speed ratios. As a result, the global constraint on the speed ratio may only be tested when calculations for every stage have been finished.
- In the same way, the *global objective function* which is, for example, the total volume of mechanical parts, is also calculated piece by piece, each stage giving its contribution to the total value.
- *Non-interference* constraints between all the objects are automatically created from the object list given by the user (Fig. 5). An interference detection criterion was programmed in order to detect every type of spatial interference between cylinders, cones, truncated cones or rings. It is applied to each couple of objects for avoiding inter-penetration or even contact.

3.4. Interactive resolution

Before starting resolution, the user enters the list of variable values (Fig. 7). Two buttons respectively permit to read or to write variables to the CAD model, which is rather convenient for interactively finding good *initial conditions* before starting optimization.

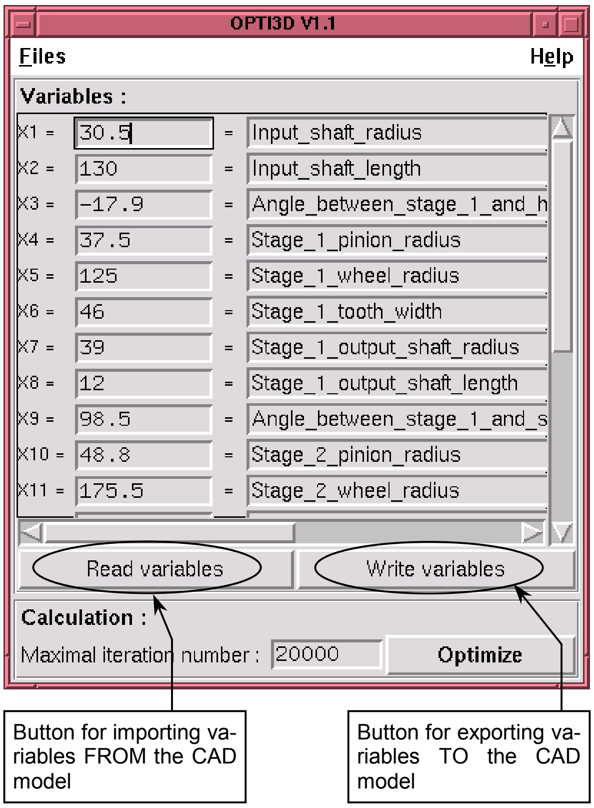

Fig. 7 : List of problem variable values.

Solving is performed in real time and all the optimizer iterations are stored and may be replayed in a sort of virtual video tape recorder (Fig. 8). Data is stored in a file where each line contains the variable vector of the corresponding iteration. The amount of data is not so big : for the example of Fig. 1, a typical file contains several thousands of lines, each of them containing 20 floating point values corresponding to the 20 variables of the CAD model. As the CAD model is parametered, it is quite easy to reconstruct it for each set of variables.

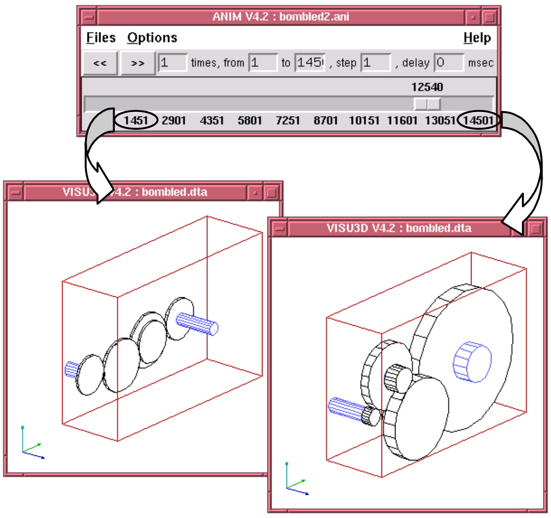

Fig. 8 : All the calculation iterations are stored and can be accessed to through a sort of virtual video tape recorder.

At any time, the user may stop the process, change variables, and so *branch out* towards a new searching direction. He may watch problem convergence from a graphical and intuitive point of view. Thus, the closing condition is far more easy to understand *graphically* (output shaft with a good position and orientation) than *numerically* (heavy trigonometric formulas).

Finally, results are explicitly displayed (Fig. 9) : a color code indicates verified constraints (green color) and violated cons-traint (red color). An intermediate orange color is used for showing still active constraints. It is useful for equality constraints which are often slightly unsatisfied at the end of the process but stay in the numerical tolerance interval. Calcu-lation time may vary, depending on the type of computer. The presented example was solved in 14501 iterations and about half an hour on a Pentium Pro 233 MHz running Linux.

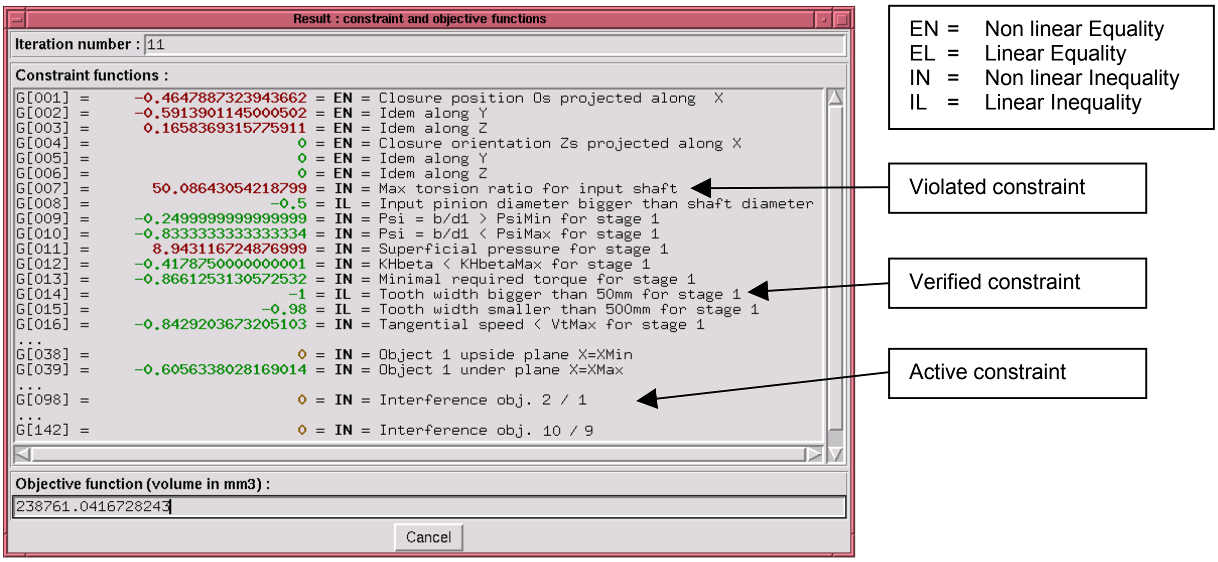

Fig. 9 : Example of objective and constraint functions display after only 11 iterations (in order to have some violated constraints)
There are 142 constraints in this problem.

4. Conclusion

A method for coupling optimizing and CAD tools was presented. It permits what we call "Collaborative Optimization", that is to say a good task repartition between both tools. This process suits perfectly the automatic resolution of a whole class of three-dimensional complex problems. It prevents the designer from writing specific optimization programs for each of the problems of the class.

The CAD tool calculates the coordinates of some critical points of the model and transmits them to the optimizer. The latter calculates better dimensions for the model, sends them back to the CAD tool and triggers a parameter update.

This original formulation might generate a little drawback : optimization time is often slow because of the CAD tool. But it is highly compensated by great advantages such as the decrease of programming time, the generic expression of the problem and the improvement of interactivity. Obviously, this method provides a more natural way of performing optimization within a CAD context.

References

[1] VANDERPLAATS G.N., " Structural optimization : where we've been and where we are going ", Proceedings of OPTI'97 (Computer aided optimum design of structures), pp. 45-54, Rome, 8th-10th Sept. 1997.

[2] "GENESIS 5.0 Fact Sheet", Vanderplaats Research & Development (VR&D), Inc., http://www.vrand.com/genesis_fact.htm

[3] "HyperOpt and OptiStruct product description", Altair Engineering, Inc., http://www.altair.com/Products

[4] FAUROUX J.C., SARTOR M., " Conception optimale de structures cinématiques 3D. Application aux mécanismes de transmission en rotation ", Proceedings of the 6th PRIMECA seminar, La Plagne, France, 7th-9th Apr. 1999.

[5] " *ADS, a FORTRAN program for Automated Design Synthesis ",* Reference manual V4.20, Engineering Design Optimization, Inc., 1987.

[6] " *DOC (Design Optimization Control) – DOT (Design Optimization Tools) ".* Reference manuals, Vanderplaats Research & Development (VR&D), Inc., 1995, http://www.vrand.com/
And also for more info on the DOC Graphical User Interface called Visual DOC, see : http://www.vrand.com/visualdocinfo.htm

[7] DATSERIS P., "Weight minimization of a speed reducer by heuristic and decomposition techniques", Mechanism & Machine Theory, 17(4):255-262, 1982.

[8] AZARM S., LI W.C., "Multi-level design optimization using global monotonicity analysis", Journal of Mechanisms, Transmissions and Automation in Design, 111:259-263, Jun. 1989.

[9] CAROLL R.K., JOHNSON G.E., "Dimensionless solution to the optimal solution of spur gear sets", Journal of Mechanical Design, 111:290-296, Jun. 1989.

[10] MICHELENA N.F., PAPALAMBROS P.Y., "Optimal model-based decomposition of powertrain system design", Journal of Mechanical Design, 117:499-505, Dec. 1995.

[11] FAUROUX J.C., SARTOR M., " Geometrical definition of a speed reducer skeleton ". Proceedings of IDMME'96, vol. 2, pp. 787-796, Ecole Centrale de Nantes, 15th -17th Apr. 1996.

[12] FAUROUX J.C., SANCHEZ C., SARTOR M., MARTINS C., " Application of a fuzzy logic ordering method to preliminary mechanism design". Proceedings of IDMME'98, vol. 2, pp. 423-430, Université de Technologie de Compiegne, 27th-29th May 1998.

[13] FAUROUX J.C., " Conception optimale de structures cinématiques tridimensionnelles. Application aux mécanismes de transmission en rotation ", PHD thesis, Laboratoire de Génie Mécanique de Toulouse, 230p., INSA Toulouse, jan. 1999. Downloadable in Postcript format at : "http://www.meca.insa-tlse.fr/~fauroux /rechera.htm".