©1996, CIVIL-COMP Ltd., Edinburgh, Scotland
B.H.V. Topping, (Editor),
Advances in Structural Engineering Optimization,
Civil-Comp Press, Edinburgh, 23 - 29

DISSENY: AN INTEGRATED SYSTEM FOR THE STRUCTURES AND STRUCTURAL ELEMENTS OPTIMAL DESIGN

P. Marti-Montrull* and P. Company-Calleja*
***Department of Mechanical Engineering, University of Murcia, Cartagena, Spain**
***Department of Technology, Jaume I University, Castellon, Spain**

Abstract

Some powerful analysis systems nowadays existing in the market, include very elemental redesign facilities. There are also more complex and sophisticated optimal design systems, but they require very expensive inversions, as much in hardware as in software. In the contrary, there is still a lack of low cost, easy-to-use tool for the independent industrial designers, and for the small industries, to abandon the craftsman trial and error method, based on synthesis-through-experience, reinforced with the knowledge of the structural response (obtained through the very well stabilized analysis processors).

In the present paper, two of the most critical aspects in the problem of the optimal design lack of acceptance (robustness and friendliness) are studied, and the solutions adopted in the optimal design of structures system DISSENY, being developed by the authors, are explained.

1 Introduction

All aspects of the development of structures and structural elements design systems, can be considered in three components: a) the data base; b) interactive communication with the user, and c) the application programs library (including finite-element based analysis methods, optimization algorithms, etc.). Efficient solutions for all three areas, independently considered, have been obtained, and some systems encompassing the whole process, whose objective is solving practical and general purpose problems, have already been developed. But such systems have not gotten the expected acceptance. Reasons for this rejection seem not being unique, and the problem was already defined in the question asked in the NATO Advanced Study Institute on Computer Aided Optimal Design, whose acts are in [1]. The question was: " Why we have problems for the designers to accept, and use, optimization methods?".

Now, CAD technologies, in its wider sense of Computer Aided Design, are firmly stabilized in the structures design systems. Such technologies include all aspects that simplify the use of the computer in the design. One of the most important aspects is the definition and implementation of specific Data Bases for the design of structures. A second important aspect is the improvement of interactive communication between user and design system, by means of the appropriate Graphical User Interface (GUI).

In spite of this, the field of Data Bases extending over the whole life cycle is now in full activity [2]. In the other hand, graphical libraries, on top of which design systems have been developed, have experimented great evolution. Consequently, optimal design systems have come to "suffer" one or more *migrations*, as data bases and graphical standards evolved. In summary, the search for fully compatible, long duration standards is still a field of interest.

In reference to the application programs, and in particular the optimization algorithms, it must be emphasized that "academic" applications tend to consider *efficiency* over any other consideration. But for the optimization to be broadly used out of the academic sphere, what must be emphasized is *robustness*. It must be considered that the efficiency of optimization algorithms is reached by tuning a great variety of parameters controlling its flow. But the meaning, the range and the more usual values for those parameters are only known by very expertise users. Moreover, assigning not proper values to some of those parameters usually imply not only a drastic decrease in the efficiency of algorithms, but even an ending by error condition. Getting an easy-to-use algorithm, implies defining default values that helps, to designers not experienced in optimal design, in making safe designs. Besides, it must allow a friendly interaction that leads the expertise user in the modification and tuning of those parameters from which the designer knows values that make a more efficient resolution for a particular case.

In addition, as no general algorithm (able to solve all kind of problem) can be obtained, we appeal to different classes of algorithms, changing from one another both while the design is still in progress, or in the beginning, depending on the kind of design. That strategy increases the need to get more robust algorithms, and a more consistent data base, and to improve the friendliness of the system, to inform, guide and suggest the user about safe but more efficient optimization alternatives. In short, we tend to establish one library of applications, including different optimization algorithms, that can be alternatively used, depending on the characteristics of both, the problem or the progress of its resolution process.

As we have seen, improving robustness in the optimization process can be done by making it easy to choose appropriate values for the parameters that control the process. In the same way, developing new design oriented models, instead of analysis oriented ones, is a very important way to improve friendliness, or user interaction (also implies a better operation of the optimization algorithm). It was Schmit [3], who first pointed the need of distinguishing between design and analysis models, concluding that design model must be placed over the analysis model in the problem formulations hierarchy. This problem must be defined in terms of a small group of "natural" design variables, very close to the designer thinking way, and paying attention to the whole manufacturing process. Nevertheless, this field is only beginning to be explored.

Next, we are going to study robustness and friendliness problems in practical application of optimal design, and we shall comment solutions adopted in DISSENY system. It is general purpose structures and structural elements optimal design system [4]. It is based in the finite elements structural analysis program ADEF (also developd by the authors [5]), in two non linear mathematical programming algorithms, and in some processors carrying the finite elements-optimization techniques coupling [6]. It has, in addition, some different modules to solve specific problems, such as basement optimization for electrical transmission towers [7], thin-wall profiles optimization [8], optimal design under non linear behavior [9], etc.

Finally, to solve all different graphical representations presents in a design system (basically: pre-processing, monitoring and post-processing), while maintaining portability and low cost, in DISSENY we opted for developing our own high level library (to manage menus and dialogues, visualize 3D models, and represent functions) [10]. To maintain independence in the high level graphical library, while easing system portability, we maintained GKS syntax. The call to the low level library from the high level one, is done by different translators, from GKS syntax to real operations in the actual low level library (for instance, Fortran PowerStation's GRAPHICS, in the Personal Computer version). This solution implies a reasonable development cost, gives the system an acceptable variety of drivers and simplifies migrations from machine, compilers or other software.

2 The Optimal Design Problem

The following is the most usual algebraic formulation for the general optimal design of structures and structural elements:

Find a design variables vector $\mathbf{x}$ $(x_1, x_2, ..., x_n)$ to:
minimize the objective function $f(\mathbf{x})$
satisfying the constraints:

$$h_j(\mathbf{x}) = 0 \quad j=1, 2, ..., m_i$$
$$g_j(\mathbf{x}) ?\ 0 \quad j=1, 2, ..., m_a \qquad (1)$$
$$x^I_i \bullet x_i \bullet x^S_i \quad i= 1, 2, ..., n$$

Where $\mathbf{x}$ is the design variables n-dimensional vector; $f(\mathbf{x})$ is the objective function; $h_j(\mathbf{x})$ is the number j of the equality constraints; $g_j(\mathbf{x})$ is the number j of the inequality constraints; x^I_i is the lower limit of variable number i, and x^S_i is the upper limit of variable number i.

Usually, the objective function $f(\mathbf{x})$ and the equality $h_j(\mathbf{x})$ and inequality $g_j(\mathbf{x})$ constraints are non linear functions. Then the problem is said to be *non linear* optimization.

Essentially, two different approaches are distinguished in optimization problem resolution: a) optimality criterion, and b) mathematical programming. The optimality criteria are methods of solving definite optimization problems, guiding the resolution by way of criteria that are known (or are believed) to fit the problem at hand. Some optimality criteria have a clear physical meaning; This is the case in the Fully Stressed Design ("F.S.D."), which is the situation in that every component is under limit stress, at least in one specified load hypothesis.

F.S.D. is a stress design of component properties. Initially, any other kind of constraint is not considered in the method, but it can be spread to include displacement constraints. The most emphasized characteristic in F.S.D. is the objective function absence. This means no merit function subject to maximize or minimize exists. That is, we cannot ensure an F.S.D. algorithm to converge, for instance, to minimum weight solution. This is because this aim cannot be explicitly fixed, if no objective function exists. Nevertheless, one good characteristic of the method is that some of its approximations get one solution very close to optimum just in the first or second iteration, with independence of initial point, making it a good method to rapidly obtain an appropriate initial design to use as starting point for more sophisticated methods.

Mathematical programming can obtain design problem solutions, applying numerical methods of constrained objective function minimization (or maximization). Many methods of numerical minimization can be defined through the same general formulation, where successive refined approaches are obtained from the previous one in the form:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + ?^k\, \mathbf{d}^k \qquad (2)$$

This means that the point of design (vector of variables $\mathbf{x}$) in the k+1 iteration, is obtained from the previous point of design, moving along a search direction ($\mathbf{d}^k$), during one certain longitude $?^k$. The main disadvantage of this search method is that a local minimum is generally obtained (a point such as no other point around it leads to a better design), which shall only be equivalent to absolute minimum in a purely casual way, or in the particular case of design space being convex. Definitely, a good solution, but not the best, is generally obtained, and, in addition, the solution obtained depends on the departure point.

Recursive quadratic programming (RQP) is a constrained mathematical programming method. In essence, the method is based on reducing the problem, in every iteration of the optimum search, to a quadratic subproblem. The basic idea was developed by Wilson [11]. Later, one algorithm based on this technique was developed by Han [12], and was implemented by Powell [13] with some modifications. RQP methods have good acceptance, mainly because: a) it has been proved they converge globally (from any arbitrary starting point), and b) they have linear (or even better) convergence ratio.

Constrained minimization algorithms require sensibility information. In particular, the ones above referenced, they need first derivatives, with respect to design variables, for the objective function and every constraint. Sensibilities can be calculated by analytical means (expensive in programming effort), or by finite differences (expensive in calculation time). Therefore the approximate-subproblems technique [14], was considered as an alternative. Even known it requires the selection and tuning of different kind of approximations to calculate estimations of all functions and its corresponding derivatives (see [15, 16], etc.).

System DISSENY employs RQP optimization methods. In concrete one implementation of Powell algorithm, and another one of Schittkowski [SCHI83] are used. The system also implements methods based on F.S.D., like Stress-Ratio, which permit obtaining a reasonably close to optimum design, to use as starting point to improve the mathematical programming algorithms' operation. Problems can be solved both by direct methods, or like succession of quadratic subproblems (using different classes of approximations).

The system uses a configuration file, which allows default values definition for all parameters controlling the selection and adjust of optimization options. In this way, a safe behavior of the system, when used by non expertise users, is guaranteed.

Defining different configuration files are allowed, so default parameters can be personalized. In addition, the default configuration can be interactively modified during program execution. Modification is done using dialogue boxes ("widgets"), that guide the user in the alternative parameters selection. In this way, expert users are allowed to improve the efficiency of the process, without damage for robustness.

3 Design Model

When a previous level is added to analysis model, considering all different classes of variables that can be defined in the design of structures (geometrical properties of components, structure geometry, material properties and structure topology), we are approaching to "design oriented model" (i.e. [18]) (figure 1). It is important to mention that, in this model, analysis information would become completely subordinated to design information, hypothetically coming to be automatically defined from design model.

It must be emphasized that, not all the four classes of variables above identified can be treated. Techniques existing in the present moment only allow effective consideration of design under geometrical properties of elements, and structure geometry. In the former, profile areas in truss or beam structures are generally considered. In the latter, variables are related to node coordinates or to variable contours. Consequently, it is habitual to consider fixed values for material properties and structure topology.

In system DISSENY, both properties and geometry variables can be considered (together or in a separate way). Besides, in case of optimization of profile areas in truss or beam structures, some techniques to obtain discrete solutions are applied. In this way, continuous solutions, given by mathematical programming algorithms being employed, are converted to commercial discrete values.

Among all aspects in the design model that have a special transcendence in the optimization process, techniques to reduce the size of the problem must be emphasized. Those techniques basically involve linking variables [19] and properties [20], and reducing the number of active constraints.

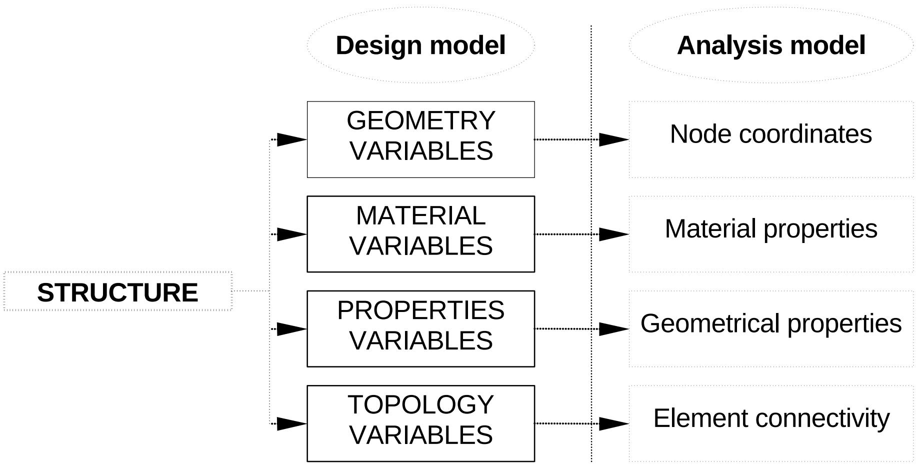

Figure 1. Design and analysis models

Initial reasons justifying the use of those techniques were the existence of practical problems to cope with more constraint and more variables than those that mathematical algorithms can handle. Nevertheless, linking of properties and geometry variables not only reduces the size of optimization problem, but, in the same time, allows introducing practical design conditions, such as geometrical ones (symmetry, proportionality, etc.), or constructive ones (simplifying assembling, limiting profile variety, etc.). So, every element property P^E can be expressed as a function of a set of properties (vector $\mathbf{P}^I$) taken as independent, in the way:

$$P^E = P^{OE} + \mathbf{T}^{EI} \cdot \mathbf{P}^I \qquad (3)$$

In the simplest case, of one group of elements sharing the same property, constant value P^{OE} is null, and all components in the vector of coefficients $\mathbf{T}^{EI}$ equal to zero, except one that equals to one.

Reduction of the number of properties needed to define a cross section, is based in mutual relations among all profiles in the same series. Operative way consists in adopting one characteristic of the section as independent (in truss and beam structures area of cross section is a good choice), and relating it with the rest of properties in the following form:

$$D = a \cdot P^b \quad (4)$$

Where D is the dependent property, P is the independent one, and parameters a and b are obtained by fitting (using least square methods) all profiles of the series to be employed.

Reducing the number of active constraints implies taking in consideration, in every iteration (or sub-problem), only those constraints that are being violated, or that have many possibilities to become “active” (violated or having a value very close to the limit) in next iterations. This is why a minimum value for the constraints is considered, to select only those having possibilities to turn active in next iterations (trying in this way to minimize changes in the subset of active constraints).

At last, and given that values of properties and geometry variables are usually very different, we use scaling techniques for variables, to improve the problem conditioning (and increase robustness).

We shall comment in more detail the distinction between leaving variables node coordinates, or defining variable contours. In one hand, in the particular case of truss or beam structures, selection of appropriate profile is relied on property variables. Consequently, if we also control node coordinates through geometry variables, all geometrical aspects are simultaneously considered. In the other hand, in a finite-elements model, coordinates of all nodes in the mesh determine not only geometry of every finite element, but global shape of the whole structure. Consequently, using node coordinates like design variables in continuous structures appeared in a natural way [21].

Nevertheless, formulating the optimization problem in terms of position of nodes in the finite element mesh, have many disadvantages when applied to continuous structures, because most design variables are required to obtain good precision in the analysis. Moreover, the obtained solutions cannot be build, because contours are discontinuous; the optimization process becomes complex, because it is necessary to regenerate the mesh every time design variables are modified, etc.

To alleviate all those drawbacks, Bennet and Botkin [22], proposed the definition of geometry by means of a succession of border elements, associated to leader points, which are, in turn, defined from a pre-defined number of design variables. Rajan [23], used straight lines and arcs of circumferences as border lines, the former defined by its extreme coordinates, the latter using the radius and initial and final angles like definition parameters. The rest of the contour curves can be defined through Bezier or B-spline curves, associating design variables to control nodes. The method of “design elements” [24, 25, 26] is used to update the mesh during optimization process. The method is based in decomposing the domain in a group of simplest geometry sub-regions, called super-elements. The mesh process on those super-elements is done in the analysis stage, intensifying the distinction between design model and analysis model.

Definition of design model for continuous structures in DISSENY is based on decomposition of the two-dimensional structural element in “design super-elements”. Those are areas with simple contours, which can be easily defined using different curve types (straight lines, circumference arcs, B-splines, etc.). An automatic mesh generator, based on mapping techniques, makes the subsequent analysis model mesh [27]. During optimization process, curves defining the contour, are changed due to modifications in those design variables associated with curve definition parameters. This contour modification implies a change in shape of super-elements, and, consequently, an update in the finite-elements mesh [28].

4 Examples

In the first example, we consider optimization of a connecting rod head (with an 7 mm. constant thickness). Assuming symmetry in the problem, model has been simplified, considering only one half of the head. Displacements are

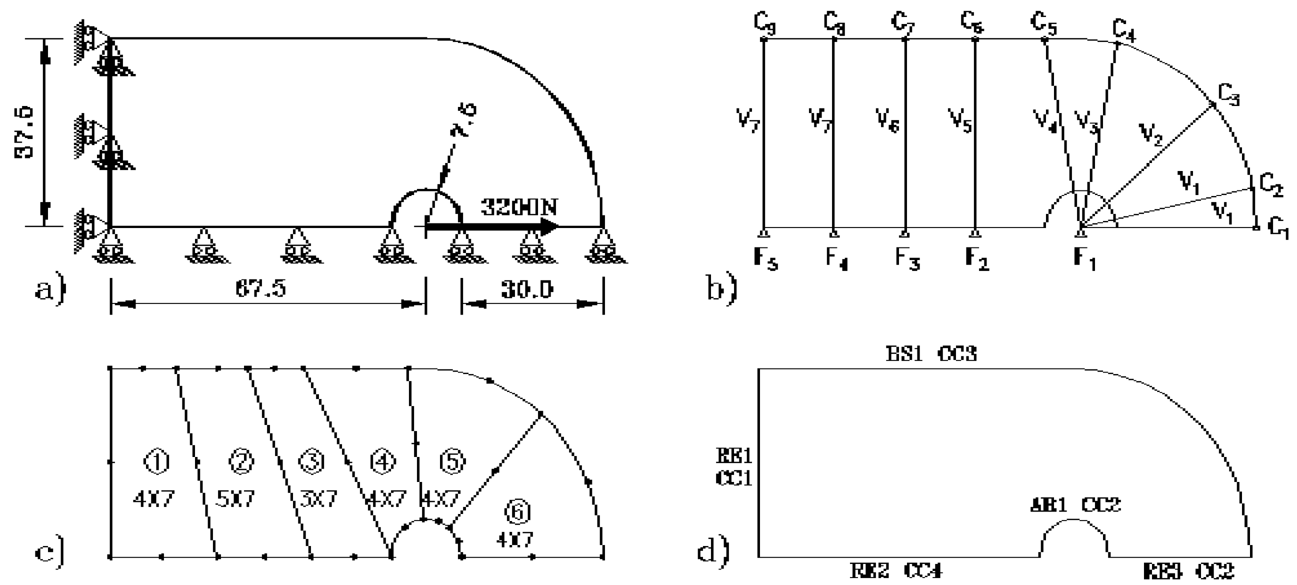

Figure 2. Rod head model. a) Dimensions, loads and restricted degrees of freedom; b) Control points, fixed points and design variables; c) Superelements and Finite Elements mesh definition; d) Contour curves

disabled, only in perpendicular direction, for nodes placed on axis. The load (3200N), is supposed to be static, and applied in the axis direction (figure 2.a). To simulate that it is applied through one bolt element fitting the hole in the head, it has been distributed according to an elliptical distribution, using Hertz equation for contact stresses.

Design variables (v_1, v_2, ..., v_7), are distances from axis to contour (figure 2.b). Variables v_1 - v_4 are measured in radial direction, with respect to the center of the hole, and v_5 - v_7 are measured normally to axis. The super-elements model consists of 6 eight-vertex super-elements (figure 2.c). When dividing super-elements, a 336 triangular finite-elements mesh, of 336 elements and 200 nodes, appears. The objective is obtaining a shape with minimum volume, and without exceeding Von Mises fluency stress (360 N/mm^2) in any element.

The design model is defined in the following way: Contour curves are made of three straight lines (RE_1, RE_2 y RE_3), one circumference arch (AR_1) and one B-Spline curve (BS_1). Tangency condition, perpendicular to symmetry axis, has been imposed in the curve-axis intersection (figure 2.d).

Six different cases have been studied, depending on the kind of the BS_1 curve. A three digit code is used to identify the case: the first digit makes reference to the degree of the curve (2 or 3), the second indicates rational (1) or non rational curve (0), and the third is 1 when curve is uniform and 0 if not. So, the case 210 corresponds to a quadratic, rational, non uniform curve.

Results of optimization (after 5 iterations), are summarized in table 1.

CASE	V_1	V_2	V_3	V_4	V_5	V_6	V_7
200	14,12	13,54	11,50	11,50	4,00	4,00	4,00
210	14,04	12,20	13,40	11,50	4,00	4,00	4,00
201	13,96	12,86	12,33	11,50	4,00	4,00	4,00
301	13,73	11,82	15,06	11,50	4,00	4,00	4,00
211	14,09	12,65	12,71	11,50	4,00	4,00	4,00
311	13,62	11,79	15,07	11,50	4,00	4,00	4,00

Table 1. Optimal values for the design variables (in mm.)

In figure 3, the evolution of objective function in all six cases is compared. And in figure 4 optimal shapes are shown.

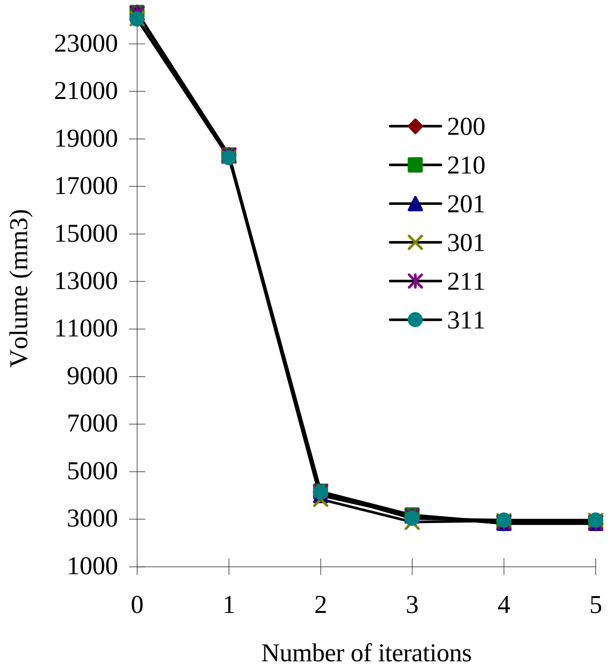

Figure 3. Evolution of the Objective Function

As can be seen in figure 3, the influence of the design model on the objective function is not important. But comparing the different shapes obtained, the influence of design model becomes apparent. It is specially remarkable than contours obtained using quadratic curves (figure 4) are better, form the point of view of smoothness, than those obtained usign cubic ones.

The second example shows the posibilities of system DISSENY in the design of complex structures. So, the image

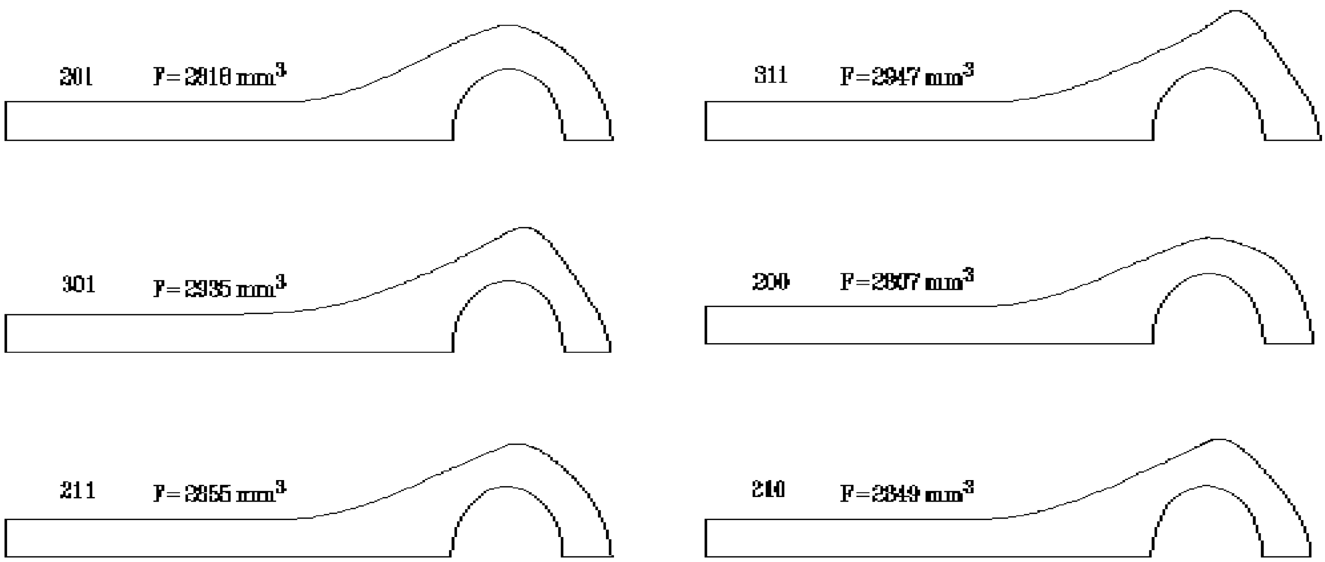

Figure 4. Optimal form and objective function values for the six cases under study

shown in figure 5, is a copy of the screen in DISSENY obtained when a session of optimal design for an electrical transmission tower has concluded. The design model appears represented in the great window in the right side. The initial tower appears in the left side of the window, and final design in the right one. Some characteristics of the GUI can be highlighted: The color map in the small window in the lower right corner is associated with final design, and shows the violation level of constraint stresses in all the bars (the color coding appears in the figure like a gray scale). Five of the windows in the left side contain representation of different evolution values associated to the process (objective function, variables of geometry, variables of properties, buckling constraints and slender constraints). The other three windows (the lower ones), display sensibilities of objective function (both the evolution and the present values) and the sensibilities of the most violated constraints, with respect to every variable. At last, and surrounding all previous windows, menu bars (up), utility menus (right), and active submenu (selection menu), are placed.

process, as much as possible. In this direction, use of “safe” algorithms is the basis. This means using algorithms, or concatenation of algorithms (i.e. FSD followed by RQP), that converge globally from any initial point, and have liner, or greater, convergence ratio. The same importance has the definition of “safe” default values for the parameters that control the optimization process. At the same time, a friendly interaction, to let the expert user tuning parameters, to obtain more fitting results in a more efficient way, was advised.

The development of design oriented models, instead of analysis oriented ones, has been justified because its absence implies an inconsistent, and inefficient data management, and even a hostile interaction. Nevertheless, the design model, in the sense of a parameterized, geometric model of the structure, has some limitations, specially when it comes to represent 3D structures. Two approaches are proposed in [29], to overcome those limitations: a) availability of different types of geometrical transformations (allowing more complex parameterizations and linkings), and b) use of “geometrical”

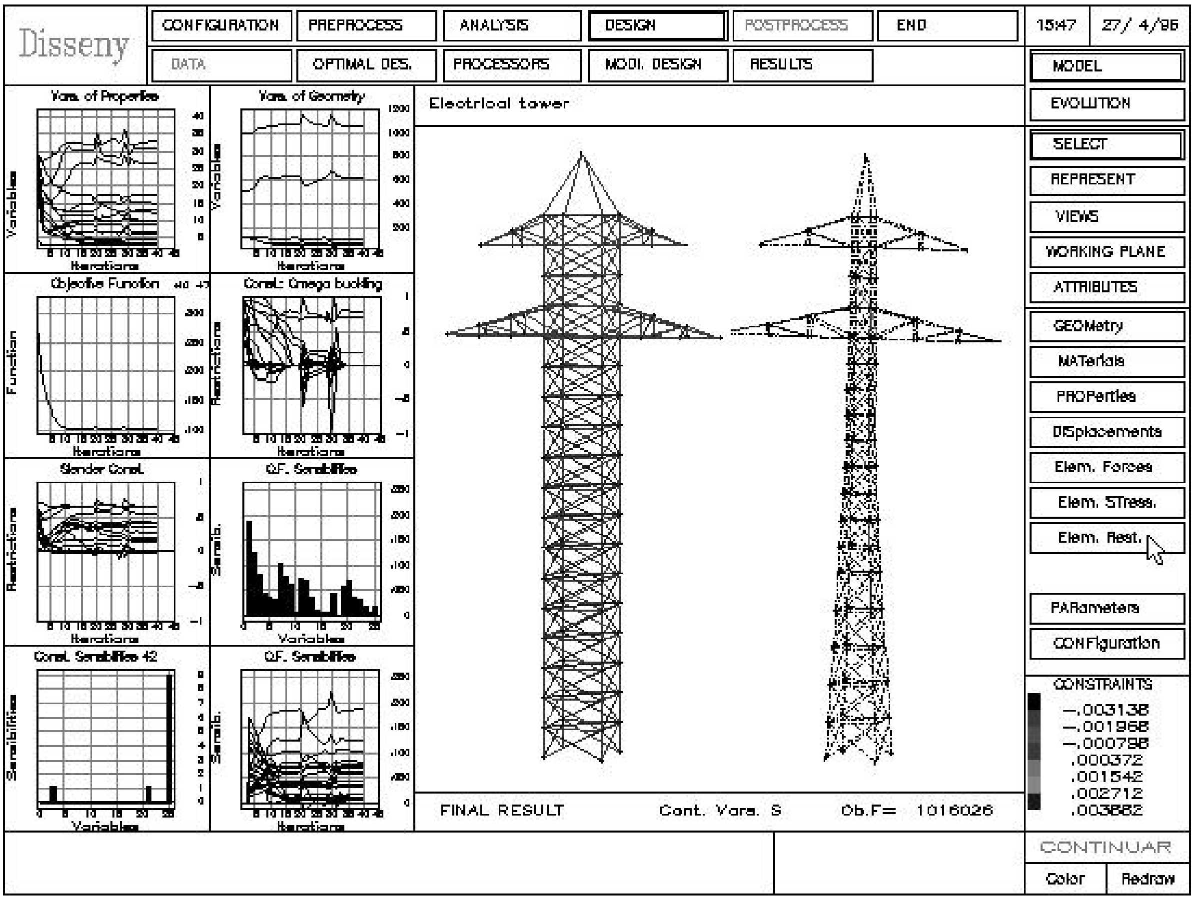

Figure 5. Screen image, during an optimal design session, in DISSENY.

models, like Constructive Solid Geometry (CSG). And we emphasize that data bases extending over the whole life cycle must be considered in this context.

The need of reducing problem sizes has been pointed. And the advantage of doing so by introducing geometrical and constructive conditions has been presented. Elements and properties linking techniques that can be used for this purpose have been explained. Scaling techniques, applied to better condition the problems, have also been described.

At last, two examples, showing possibilities of system DISSENY, have been given.

5 Conclusions

An integrated system to *assist* in design has been presented, based on optimization techniques, working in an interactive way (with a great participation of graphical representations), and created around a problem-fitting data base.

To increase easiness of use without loosing robustness, we first proposed automating and encapsulating the optimization

6 References

[1] Mota Soares, C.: (editor). (1987). *Computer Aided Optimal Design: Structural and Mechanical Systems*. Ed. Springer-Verlag, Berlin-Heidelberg 1987.

[2] Eastman, M. and Fereshetian, N. (1994). Information Models for Use in Product Design: a Comparison. *Computer-Aided Design, Vol. 26, Nº7*. 1994.

[3] Schmit, L.A. Concluding Remarks. In *The Optimum Shape Automated Structural Design*. J.A. Bennet and M.E. Botkin (Ed.). Plenum Press, 1986.

[4] Martí, P., Company, P. y Sanchis, M. (1985). DISSENY. Un sistema interactivo para el diseño de estructuras basado en técnicas de optimización. *Anales de Ingeniería Mecánica, Año 3*, nº I. 1985. p. 285-290.

[5] Martí, P. y Company, P. (1993). *ADEF. Análisis interactivo de estructuras por elementos finitos.* Ed. Universidad de Murcia, 1993.

[6] Martí, P., Company, P., y Sanchis, M. (1986). Acoplamiento elementos finitos-técnicas de optimización en el sistema DISSENY. *II Simposium sobre Aplicaciones del Método de los Elementos Finitos en Ingeniería,* Barcelona,. 1986. p. A-133 a A-147.

[7] Martí P., Ortueta C. y Company P. (1990). Optimización conjunta de torres y cimentaciones de líneas de conducción eléctrica. *Memorias del I congreso de Métodos Numéricos en Ingeniería.* Ed. G.Winter y M.Galante, Barcelona, 1990, p 725-732.

[8] García-Estañ, J. y Martí P. (1994). Optimización de estructuras metálicas con perfiles de pared delgada. *Anales de Ingeniería Mecánica, Año 10, nºI.* 1994. p. 393-400.

[9] López-Alanís, C., Sánchez, G. y Martí P. (1994). Diseño óptimo de torres de transmisión de energía eléctrica teniendo en cuenta el comportamiento no lineal (grandes desplazamientos). *Anales de Ingeniería Mecánica, Año 10, nºI.* 1994. p. 409-416,

[10] Company P. y Martí P. (1994). Portabilidad de los gráficos interactivos para el diseño de estructuras en el sistema DISSENY. *Anales de Ingeniería Mecánica, Año 10, nºI,* 1994. p. 645-652,

[11] Wilson, R.B. (1963). A Simplicial Algorithm for Concave Programming. *Ph. D. Thesis*, Graduate School of Business Administration, Hardward University, Boston, 1963.

[12] Han, S.P. (1977). A Globally Convergent Method for Nonlinear Programming. *J. Optimization Theory Appl. 22*. 1977. p. 297-309,

[13] Powell, M.J.D. (1978). A Fast Algorithm for Nonlinear Constrained Optimization Calculations. *Lecture Notes in Mathematics,* Vol. 630. Springer Verlag, 1978.

[14] Schmit, L.A. and Miura, H. (1976). Approximation Concepts for Efficient Structural Synthesis. *NASA Contracttor Rep.*, NASA-CR-2552, 1976.

[15] Prassad, B. (1983). Explicit constraint approximation forms in structural optimization. Part 1: Analysis and projections. *Computer Methods in Applied Mechanics and Engineering,* Vol. 40. 1983. p 1-26.

[16] Svanberg, K. (1987). The Method of Moving Asymptotes. A New Method for Structural Optimization. *International Journal for Numerical Methods in Engineering.,* Vol 24. 1987. p 359-373.

[17] Schittkowski ,K. (1983). On The Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function. *Math. Operationsforsch u Statist. Ser. Optimization*, 14(2). 1983. p. 197-216.

[18] Fleury, C. (1986). Computer Aided Optimal Design of Elastic Structures. En: Mota Soares C.: editor. *Computer Aided Optimal Design: Structural and Mechanical Systems*. Ed. Springer-Verlag, Berlin-Heidelberg 1986. pp 831-900.

[19] Martí, P., Company, P. y Mas F. (1986). Diseño optimo de estructuras espaciales de nudos articulados con geometría variable. *Anales de Ingeniería Mecánica, No. 4(2).* 1986. p. 213-218.

[20] Martí P. y Company P. (1987). Optimización de mallas espaciales de nudos articulados. *Anales de Ingeniería Mecánica,* nº *5(2),* p. 147-152, 1987.

[21] Zienkiewicz, O.C. and Campbell, J.S. (1973). Shape Optimization and Sequential Linear Programming. In: Gallagher, R.H. y Zienkiewicz, O.C. (Ed.). *Optimum Structural Design,* Wiley, 1973. p. 109-126.

[22] Bennet, J.A. and Botkin, M.E. (1983). Shape Optimization of Two Dimensional Structures whith Geometric Problem Description and Adaptative Mesh Refinement. *AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*. Lake Tahoe, Nevada, 1983.

[23] Rajan, S.D., Belegundu, A.D. and Budiman, J. An Integrated System for Shape Optimal Design. *Computers & Structures, Vol. 30*, nº 1/2. p. 337-346.

[24] Iman, H. (1982). Three Dimensional Shape Optimization. *International Journal for Numerical Methods in Engineering, Vol. 18*. 1982. p. 661-673.

[25] Botkin, M.E. (1983). Shape Optimization of Plate and Shell Structures. *AIAA Journal, Vol 20, nº 2*. 1983. p. 268-273.

[26] Wasserman, K. (1984). On Designing a Database Managemente System for a Computer-Aided Engineering Software System. *Computer & Structures vol. 22, nº 2.* 1984. p. 205-212.

[27] George, P.L. (1991). *Automatic Mesh Generation.* Ed. John Wiley & Sons. 1991.

[28] López, D., Cañavate, F. y Martí P. (1994). Diseño óptimo de forma de elementos estructurales planos. *Anales de Ingeniería Mecánica, Año 10, nºI.* 1994. p. 401-408.

[29] Rasmussen, J.; Lund, E. and Olhoff, N. (1993). Parametric Modeling and Analysis for Optimum Design. In *Proceedings of "The World Congress on Optimal Design of Structural Systems"*, Compiled for: J. Herskovits. Vol. 2. 1993. p. 407-414.