

From: International Journal of Solids and Structures, Vol. 26. No. 9/10, pp. 1173-1210, 1990  
(a special issue of invited papers compiled in memory of Professor Charles Babcock, CALTECH)

## **GENOPT-A PROGRAM THAT WRITES USER-FRIENDLY OPTIMIZATION CODE**

David Bushnell, Dept 93-30/251, Lockheed Palo Alto Research. 3251 Hanover Street, Palo Alto, CA 94304

(This is an abridged version. See the full-length paper for more: [genopt.papers/1990.basic.pdf](http://genopt.papers/1990.basic.pdf) )

### **ABSTRACT**

The purpose of GENOPT (GENeral OPTimization) is to enable an engineer to create a user-friendly system of programs for analyzing and/or optimizing anything. The application of GENOPT is not limited to the field of structural mechanics. GENOPT is designed to handle problems with small data bases, not large finite element models, although it might well be used to provide a user-friendly "shell" within which any analysis could be done. GENOPT is ideal for generating programs for optimizing objects the behavior (stress, buckling, vibration, etc.) of which can be expressed by relatively simple tables or formulas such as those that appear in handbooks, or for optimizing objects the behavior of which has been previously encoded in existing subroutines. The optimizer used in GENOPT, created by Vanderplaats, is called ADS. Two examples are presented, one relatively simple, the other more complex. **(NOTE: Since this was written GENOPT has been "hooked up" with BIGBOSOR4, an analyzer of shells of revolution and prismatic panels and shells.)**

### **INTRODUCTION**

When engineers embark on the task of designing an object that must survive certain environments during service, they often develop or use computer programs that analyze given, fixed configurations. If, under the various applied loads, a configuration appears to be inadequate, certain dimensions, materials, or other parameters are changed by the engineer and the analysis is repeated. The design evolves by means of this "manual" iterative process, which may require the expenditure of much labor and the passage of many days. Should a certain configuration prove satisfactory, that is, should the analysis program or programs show that this configuration survives all the environments with adequate margins of safety, the engineer is sorely tempted to terminate the iteration process and accept the design even though, while feasible, it may not be optimum.

The step from simple analysis to automated optimization seems to be a difficult one for many engineers and their managers. In every field there are many proven computer programs that analyze things with given configurations and given environments. Therefore, workers in a field are willing to accept the results of computerized analyses and to use these results as a basis for decisions on what to do next. However, these engineers and managers often seem less willing at present to allow the computer to make decisions about how the dimensions and other properties of the configuration are to be changed in order to minimize cost, minimize weight, or meet some other objective. Entering the field of automated optimization seems scary and appears to require mathematical expertise not available in time to meet the goals of a design project.

The main purpose of GENOPT is to make this step into the world of automated optimization easy. If the engineer has formulas from handbooks such as Roark's (1954), design curves, and/or algorithms for predicting

the behavior of given configurations, GENOPT, working with these analysis tools, will generate a program system that can find the "best" design in a user-friendly way.

The user of GENOPT does not need to know much about optimization as a mathematical discipline. The optimizer used in GENOPT is called ADS. It was created by Vanderplaats (Vanderplaats and Sugimoto, 1986; Vanderplaats, 1987). ADS is "hardwired" in the "0-5-7" mode, which is the reliable "modified-method-of-feasible-directions" branch of this widely used optimization software.

The user of GENOPT must establish:

- (1) what types of behavior should be included in the analysis?
- (2) what allowable limits should be assigned to each type of behavior?
- (3) what factor of safety should be assigned to each type of behavior?
- (4) what is the objective of the optimization?

Over the past several years the author has developed several program systems written in FORTRAN that allow the user to analyze and optimize certain kinds of structures (Bushnell, 1983a, b, 1986a, b, 1987, 1988). These program systems have a similar style and structure. In a preprocessor, usually called "BEGIN", the user is asked in an interactive mode to supply a starting design, material properties, boundary conditions and loads. In a second processor called 'DECIDE' the user is asked in an interactive mode to choose decision variables, lower and upper bounds of decision variables, linked variables, and something called "escape" variables (variables that are iteratively increased in size by 10% per iteration until optimization via ADS takes over). Then in a third processor called "MAINSETUP" the user is asked in an interactive mode to provide certain strategy parameters for the batch-mode analysis to follow. Finally, the user launches a batch run involving a mainprocessor. If the problem is an optimization problem, this batch run will yield a new design. Three of the program systems (Bushnell, 1983a, b, 1987, 1988) have a "CHANGE" processor which permits the user to assign new values to certain of the variables without having to run a case from "BEGIN". The three program systems (Bushnell, 1983a, b, 1987, 1988; have in common a user-friendly feature that definitions of the data appear throughout the output so that this output is easy to understand.

The idea is close at hand to offer the engineer the potential for developing analysis optimization program systems that have a similar user-friendly style and structure. It should be possible to write a "master" computer program that, eliciting certain information from its user, generates program systems that perform tasks analogous to those performed by the program systems described in Bushnell (1983a, b, 1987, 1988) tasks that are defined by the user of this "master" program. This paper describes the implementation of this idea.

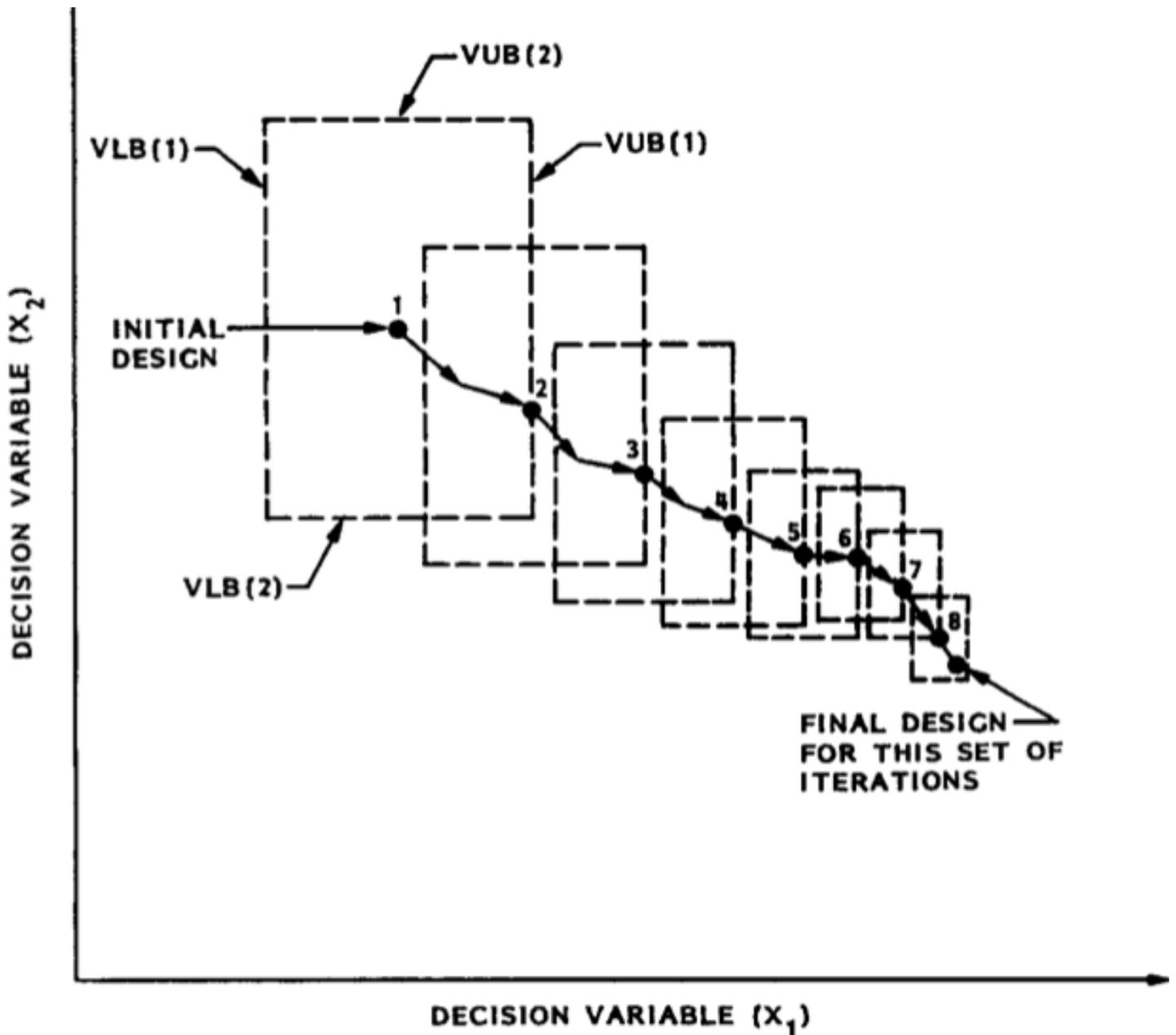


Fig. 2 Schematic of the evolution of a design with one set of design iterations and two decision variables,  $X(1)$  and  $X(2)$ . With each iteration the optimizer, ADS, establishes a “window” of permitted excursion of the decision variables. In GENOPT this “window” shrinks by a factor of 0.8 with each design iteration. Upon re-execution of OPTIMIZE the “window” is re-expanded to its original size, which depends upon lower and upper bounds supplied by the end user in DECIDE and upon certain strategies used by ADS. (from International Journal of Solids and Structures, Vol. 26. No. 9/10, pp. 1173-1210, 1990)

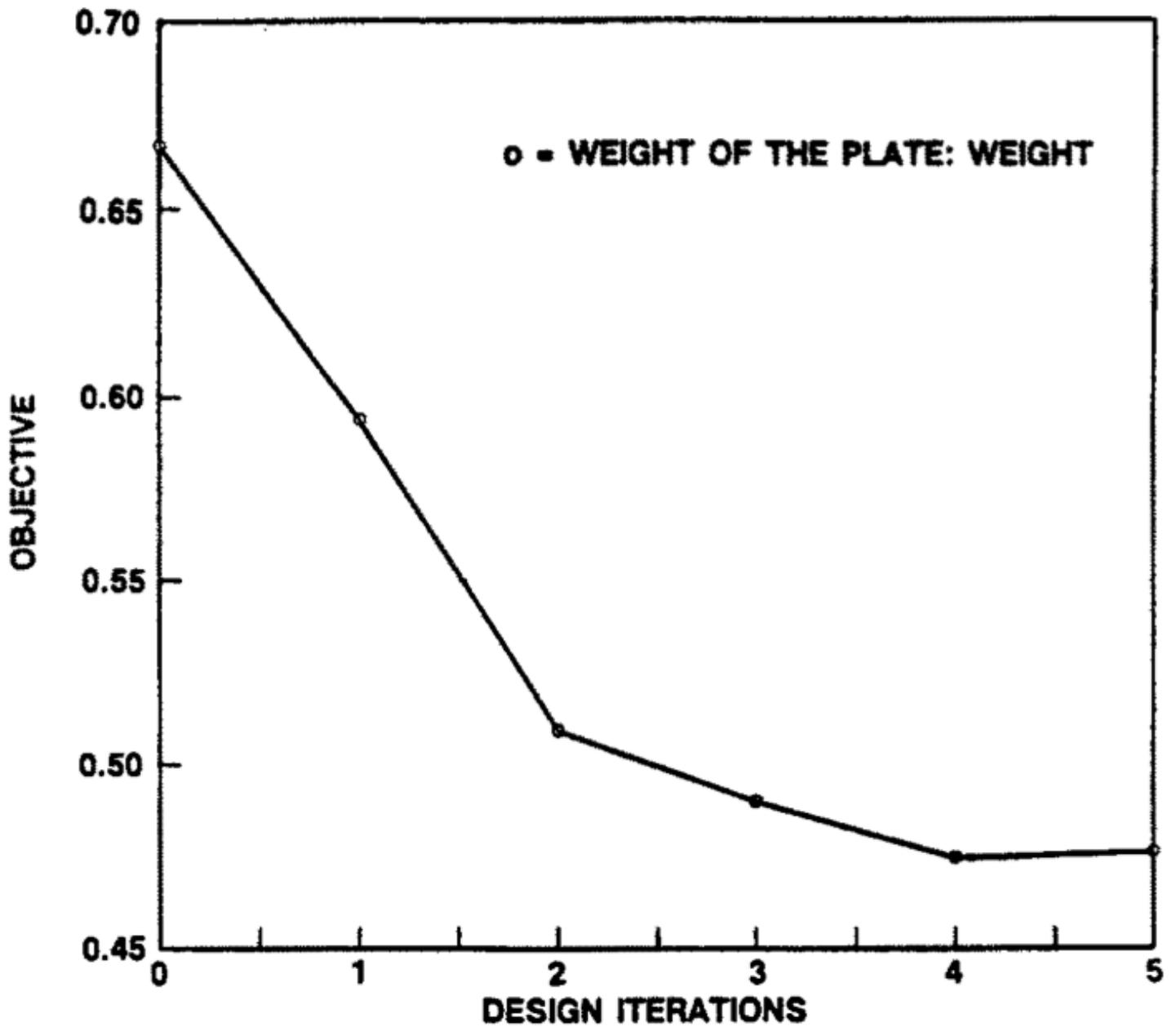


Fig. 3 Minimum-weight design of a rectangular, isotropic plate versus design iterations. (from International Journal of Solids and Structures, Vol. 26. No. 9/10, pp. 1173-1210, 1990)

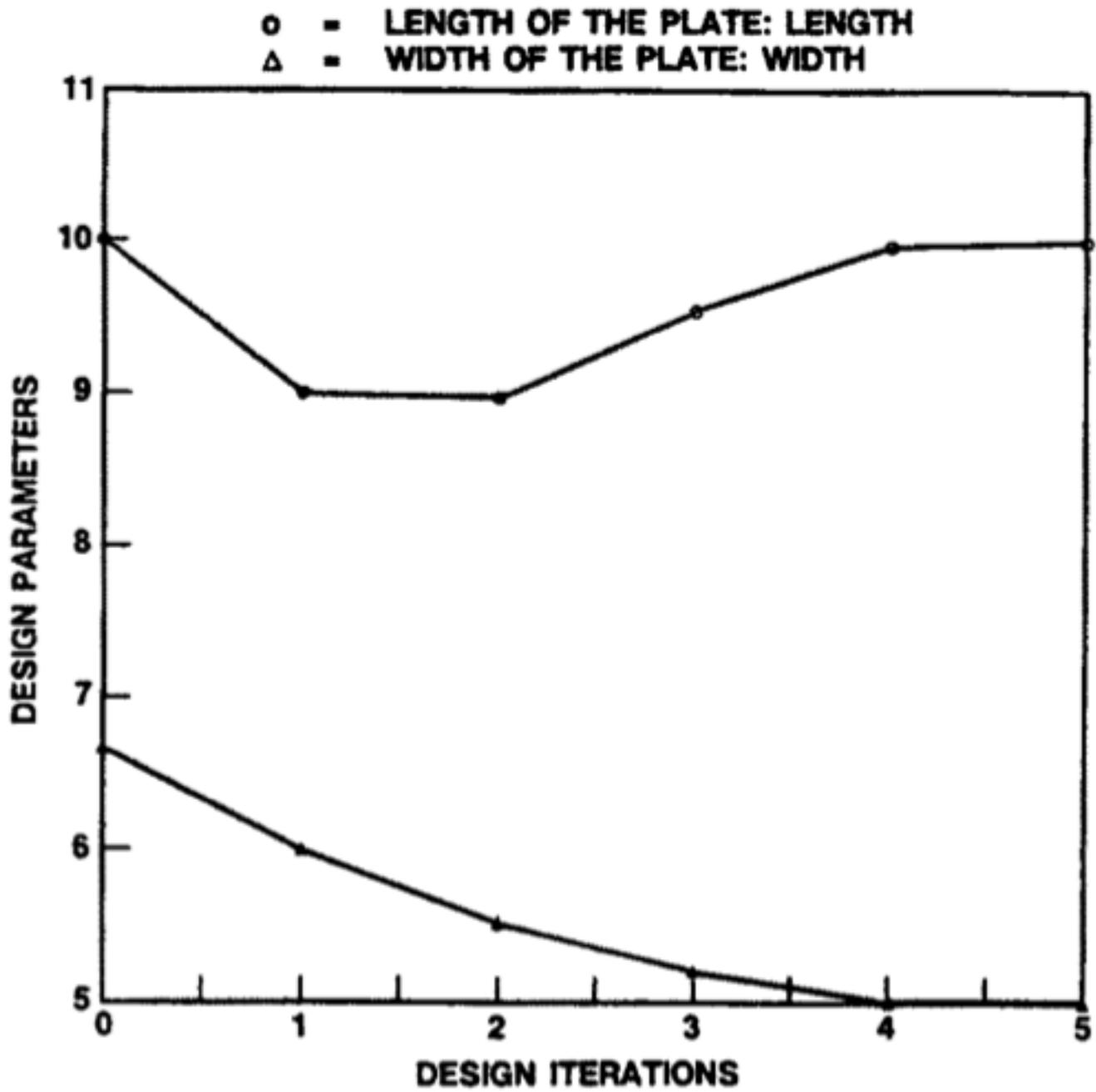


Fig. 4 Minimum-weight design of the plate. Decision variables LENGTH and WIDTH versus design iterations. (from International Journal of Solids and Structures, Vol. 26, No. 9/10, pp. 1173-1210, 1990)

- =  $1. - (1.00 - 1.00 \cdot \text{VAR}(2))^{**}(1.00) + 50.0 \cdot \text{VAR}(3)^{**}(-1.00)$
- △ =  $\text{MAXSTR}(1) / (\text{STRESS}(1) \times \text{FSTRES}(1)) - 1$ ; F.S. = 1.10
- + =  $\text{BUCKLE}(1) / (\text{MINBUC}(1) \times \text{FBUCKL}(1)) - 1$ ; F.S. = 1.20
- × =  $\text{MAXSTR}(2) / (\text{STRESS}(2) \times \text{FSTRES}(2)) - 1$ ; F.S. = 1.10
- ◇ =  $\text{BUCKLE}(2) / (\text{MINBUC}(2) \times \text{FBUCKL}(2)) - 1$ ; F.S. = 1.20
- ▽ =  $\text{MAXSTR}(3) / (\text{STRESS}(3) \times \text{FSTRES}(3)) - 1$ ; F.S. = 1.00
- ⊠ =  $\text{FREQ}(3) / (\text{MINCPS}(3) \times \text{FSFREQ}(3)) - 1$ ; F.S. = 1.00
- \* =  $\text{AW}(3) / (\text{W}(3) \times \text{FW}(3)) - 1$ ; F.S. = 1.00

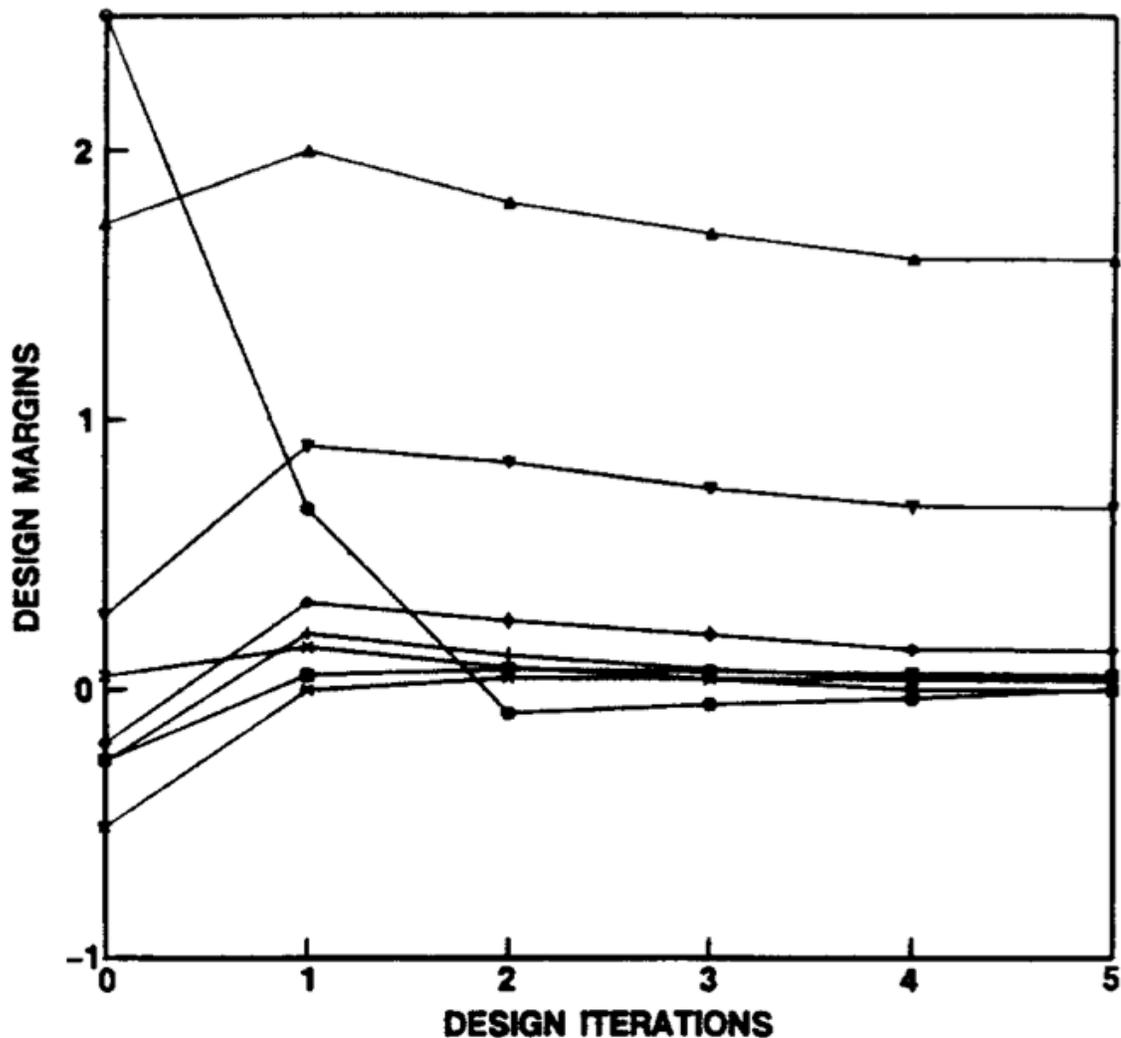


Fig. 5 Minimum-weight design of the plate. History of the design margins versus design iterations. The quantities in the legend are defined as follows: VAR(2), VAR(3) = LENGTH, WIDTH (see Table 8); the remaining variables (MAXSTR, STRESS, FSTRESS, BUCKLE, MINBUC, FBUCKL, etc.) are defined in Table 2. F.S. = "factor of safety". (from International Journal of Solids and Structures, Vol. 26, No. 9/10, pp. 1173-1210, 1990)