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

David Bushnell

Consulting Scientist, Dept. 93-30/251, Lockheed Palo Alto Research,
3251 Hanover Street, Palo Alto, California, 94304, Tel: (415)424-3237

ABSTRACT

The purpose of GENOPT (GENeral OPTimization) is to enable an engineer to create, more or less automatically, a user-friendly system of programs for analyzing and/or optimizing something. 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 things the behavior (stress, buckling, vibration, etc.) of which can be expressed by relatively simple tables or formulas such as those that appear in Roark's handbook [1], or for optimizing things the behavior of which has been previously encoded in existing subroutines. The optimizer used in GENOPT is called CONMIN, and was written by Vanderplaats [2]. It is based on the method of feasible directions [3]. Two examples are presented, one relatively simple, the other more complex.

INTRODUCTION

When engineers embark on the task of designing something 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, 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 user does need to know:

1. what types of behavior should be included in the analysis,

2. what allowable should be assigned to each type of behavior,

3. what factor of safety should be assigned to each type of behavior,

4. what the objective of the optimization is.

Over the past several years the writer has developed several program systems written in FORTRAN that allow the user to analyze and optimize certain kinds of structures [4]-[9]. 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 ten per cent per iteration until optimization via CONMIN 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 [6-8] 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 [6-8] have in common a user-friendly feature that definitions of the data appear throughout the output so that this output is easy to understand.

Recently the idea occurred to the writer, why not make it easy for the engineer to develop analysis/optimization program systems that have a similar user-friendly style and structure? Why not 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 [6-9], tasks that are defined by the user of this ``master'' program?

This paper describes the implementation of this idea. Further details appear in [10].

GENOPT COMMANDS

```
bush-> genoptlog

GENOPT commands have been activated.

  gentext      GENOPT user generates a prompt file.
  genprograms  GENOPT user generates (makes) executables:
               begin, decide, mainsetup, optimize,
               change, chooseplot, and diplot.
  begin        End user provides starting data.
  decide       End user chooses decision variables, bounds,
               linked variables, and inequality constraints.
  mainsetup    End user sets up strategy parameters.
  optimize     End user performs optimization.
  change       End user changes some parameters.
  autochange   New values for decision variables randomly
  superopt     End user find global optimum (autochange/optimize)...
  chooseplot   End user chooses which variable to plot vs.
               iterations.
  diplot       End user plots variables vs. iterations.
  insert       GENOPT user adds parameters to the problem.
  cleangen     GENOPT user cleans up GENeric case files.
  cleanspec    End user cleans up SPECific case files.

Please consult the following sources for more about GENOPT:
  1.  genopt.story, howto.run, and genopt.news (in the
      .../genopt/doc directory)
```

2. Sample cases, plate.cas and panel.cas (in the .../genopt/case directory)
3. NAME.DEF file, where NAME is the generic case name.
4. GENOPT.HLP file   (Type "helpg".)