THE TEACHING OF CREATIVE MATHEMATICAL MODELING VIA AN EDUCATIONAL TOOLKIT FOR DESIGN OPTIMIZATION (TDO)

Jan A. SNYMAN
Department of Mechanical Engineering
University of Pretoria, 0002, South Africa
jan.snyman@eng.up.ac.za

Danie J. DE KOCK
Department of Mechanical Engineering
University of Pretoria, 0002, South Africa
djdekock@postino.up.ac.za

ABSTRACT

The solution of a real-life problem via mathematical modeling often leads to the posing of a mathematical optimization problem. Even if the modeling exercise is relatively simple, the solution of the associated optimization problem represents a non-trivial and time-consuming process. In the teaching of mathematical modeling, this fact often inhibits the student from carrying out the repetitive but essential evaluation of various alternative models in order to arrive at an acceptable solution. To overcome this difficulty, the Toolkit for Design Optimization (TDO) was recently developed (Snyman et al. 2001). This system allows the student to easily solve his or her formulated optimization problem on a computer, through the interactive use of a graphical user interface (GUI), without doing any formal programming. This paper briefly describes the system, and presents some experiences of the authors in using TDO in teaching a course in creative modeling to a group of senior undergraduate engineering students. With very little formal knowledge of mathematical optimization algorithms, the students were capable of solving a wide range of modeling problems. Of particular importance is the finding that the system not only enables the students to be creative in solving non-trivial design problems, but also allows them to have fun in doing so.

Keywords: computing technology, mathematical modeling, optimization algorithms

1. Introduction

The attempt at solving a real-life problem via mathematical modeling requires the cyclic performance of the four steps depicted in Figure 1. The main steps are: 1) the observation and study of the real-world situation associated with a practical problem, 2) the abstraction of the problem by the construction of a mathematical model that is described in terms of preliminarily fixed model parameters $\mathbf{p}$, and variables $\mathbf{x}$ that have to be determined such that model performs in an acceptable manner, 3) the solution of a resulting purely mathematical problem that requires an analytical or numerical solution $\mathbf{x}^*(\mathbf{p})$, and 4) the evaluation of the solution $\mathbf{x}^*(\mathbf{p})$ and its practical implications. After step 4) it may be necessary to adjust the parameters and to refine the model, resulting in a new mathematical problem to be solved with an associated new solution to be evaluated. It may be required to perform the modeling cycle a number of times before an acceptable solution is obtained. More often than not, the mathematical problem to be solved in 3) is a mathematical optimization problem requiring a numerical solution. In many cases, even if the modeling exercise is relatively simple, the solution of the formulated optimization problem represents a non-trivial and time-consuming process. In the teaching of mathematical modeling, this fact often inhibits the student from carrying out the repetitive but essential evaluation of various alternative models in order to arrive at a practical solution. The Toolkit for Design Optimization (TDO) (Snyman et al. 2001) allows the student to easily solve his or her formulated constrained or unconstrained optimization problem on a computer, through the interactive use of a graphical user interface (GUI) without doing any formal programming.

TDO employs gradient-based optimization algorithms and depending on the type of problem being solved the student has the option of experimenting with different algorithms. TDO can be used to select an analytical objective function to be optimized as well as additional analytical equality and inequality constraint functions if constrained problems are considered. Allowance is also made for the use of approximations in specifying the objective and constraint functions.

In this paper the use of the toolkit is illustrated through its application to two sample mathematical modeling problems, typical of those that may be posed in the classroom. The first problem is the determination of the minimum cost design of a beer can of prescribed volume. The objective of the second example is to find the equilibrium configuration of a cable of negligible weight subjected to concentrated loads. Experiences of the authors with TDO in teaching a course in creative modeling to a group of senior undergraduate engineering students are also discussed. Of particular importance is the finding that the system not only enables the students to be creative in solving non-trivial design problems, but also ensures that they have fun in doing so.

2. Statement and Numerical Solution of an Optimization Problem

A *mathematical optimization problem* can be stated as follows:

Find $\mathbf{x}=(x_1,x_2,\ldots,x_n)\in R^n$, that minimizes $f(\mathbf{x})$ subject to the constraints

$$g_j(\mathbf{x})\leq 0,\quad j=1,2,\ldots,m \text{ and} \qquad (2.1)$$
$$h_j(\mathbf{x})=0,\quad j=1,2,\ldots r$$

where $f(\mathbf{x})$, $g_j(\mathbf{x})$ and $h_j(\mathbf{x})$ are scalar functions of the *variables* $\mathbf{x}$. The function f is called the *objective function* and g_j and h_j are respectively the *inequality* and *equality constraint functions*. A *local optimum* solution is denoted by $\mathbf{x}^*$.

TDO uses gradient-based optimization methods developed at the University of Pretoria to solve the above general problem. These methods have the common and unique property that no explicit line searches are required. The individual algorithms that may be selected by the student are LFOP (Snyman 1982 and 1983), ETOP (Snyman 1985), and SQSD (Snyman and Hay 2000a) for unconstrained optimization, and LFOPC (Snyman 2000), ETOPC (Snyman 1998) and DYNAMIC-Q (Snyman et al. 1994 and Snyman and Hay 2000b) for constrained problems. With ETOP(C) both a Fletcher-Reeves or a Polak-Ribiére implementation is available. DYNAMIC-Q allows for the solution of (2.1) through the solution of a sequence of simple quadratic approximate sub-problems, constructed from the sampling of the function values and gradient values of the objective and constraint functions at successive approximate solution points.

3. Mathematical Modeling

The *formulation* of a mathematical modeling problem as an optimization problem involves transcribing a verbal description of the problem into a well defined mathematical statement by performing the following three steps (Arora 1989) (i) In addition to fixed parameters **p**, identify a set of design variables to describe the system, i.e., the n-dimensional vector $\mathbf{x}=(x_1,x_2,\ldots, x_n)$. (ii) Determine a criterion that is needed to judge whether or a given model, corresponding to a given **x** is better than another. This criterion is called the objective function f and is of course influenced by the variables, i.e., $f=f(\mathbf{x})$. (iii) Specify the set of constraints within which the system must perform. Again the specified constraints are influenced by the design variables of the system. If the design satisfies all the constraints we have a *feasible* (workable) system or model.

The following two examples are typical of simple modeling problems that may be posed in the classroom. They will later be used as vehicles to illustrate the implementation of TDO.

3.1. Beer Can Problem (Arora 1989) The verbal statement of the design problem is as follows. Design a can that will hold at least a specified amount of beer and meet other design requirements. The cans will be produced in billions, so that it is desirable to minimize the cost of manufacturing them. Since the cost can be related directly to the surface area of the sheet metal used, it is reasonable to minimize the sheet metal required to fabricate the can. Fabrication, handling and aesthetics and shipping considerations impose the following restrictions on the size of the can: 1) the diameter should not be more than 8 cm and not less than 3.5 cm; 2) the height of the can should be no more than 18 cm and no less than 8 cm; and 3) the can is required to hold at least a specified volume, V_{spec} ml, of fluid (e.g., V_{spec}=400 ml =400 cm^3). The mathematical formulation is now obtained by performing the following steps. (i) The design variables are identified as x_1=D= diameter of the can (cm) and x_2=H= height of the can (cm). (ii) The objective function to be minimized is the total surface area of the can: area = $\pi DH+\frac{1}{2}\pi D^2$. This gives $f(\mathbf{x})=\pi x_1x_2+\frac{1}{2}\pi x_1^2$. (iii) From the statement of the problem the following inequality constraints are identified. The volume = $\frac{1}{4}\pi D^2H \geq V_{spec}$, i.e. , $g(\mathbf{x})= V_{spec}-\frac{1}{4}\pi x_1^2 x_2\leq 0$. Constraints on the size can imply: $3.5\leq D = x_1 \leq 8$ and $8\leq H = x_2 \leq 18$. The final formal mathematical statement of the design optimization problem is therefore:

minimize $f(\mathbf{x})=\pi x_1x_2+\frac{1}{2}\pi x_1^2$

such that $g_1(\mathbf{x})= V_{spec}-\frac{1}{4}\pi x_1^2 x_2\leq 0$ with *side constraints* (3.1)

$$g_2(\mathbf{x}) = 3.5 - x_1 \leq 0;\, g_3(\mathbf{x}) = x_1 - 8 \leq 0;\, g_4(\mathbf{x}) = 8 - x_2 \leq 0;\; g_5(\mathbf{x}) = x_2 - 18 \leq 0$$

3.2. Cable Configuration Problem Consider the *symmetrical* system of three masses supported by an inextensible cable of negligible weight as shown in Figure 2. The problem is to find the equilibrium configurations of the cable for different choices of masses m_1 and m_2, and connecting lengths ℓ_1 and ℓ_2. This problem may also be formulated as an optimization problem by performing the following three necessary steps. (i) Identify the relevant variables as (x_3, x_1), the Cartesian coordinates of mass m_1 and x_2 the vertical position of mass m_2. (ii) Recognize, from elementary energy considerations, that for any given choice of the masses and the connecting lengths, the equilibrium configuration corresponds to that of minimum potential energy, i.e., choose the objective function as $f(\mathbf{x})= 2m_1gx_1+m_2gx_2$ or more simply, $f(\mathbf{x})=2m_1x_1+m_2x_2$ since the acceleration due to gravity g is constant. (iii) As the cable is inextensible, specify the associated constraints $x_1^2 + x_3^2 \le \ell_1^2$ and $(x_1 - x_2)^2 + (1 - x_3)^2 \le \ell_2^2$.

The final formal mathematical statement of the cable design optimization problem is therefore: minimize $f(\mathbf{x})=2m_1x_1+m_2x_2$

such that (3.2)

$$g_1(\mathbf{x}) = x_1^2 + x_3^2 - \ell_1^2 \le 0\,;\; g_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 - 2x_1x_2 - 2x_3 + 1 - \ell_2^2 \le 0$$

3.3 Modeling-Optimization Interaction In the modeling process the student would normally like to quickly evaluate different options and strategies to arrive at an acceptable practical solution. This would normally require the changing of the different *parameters* of the specific problem, and then solving the correspondingly modified optimization problem to evaluate various alternatives. In the beer can problem (3.1) the typical parameter is V_{spec} , and in the cable problem (3.2) the parameters are the masses m_1 and m_2, and connecting lengths ℓ_1 and ℓ_2. Although the modification of the model through parameter variation is simple, the solution to the resulting reformulated optimization problem may be non-trivial by comparison. The latter exercise may also be time-consuming and distracting. Therefore, if the emphasis in the classroom is to be on the modeling aspects, i.e., on the formulation and evaluation of different models, then the availability of a computational device that may easily and quickly be used to solve the different formulated optimization problems, would clearly be an invaluable aid. The TDO graphical user interface is such a computational tool.

4. Graphical User Interface

4.1. Main Window The Toolkit for Design Optimization (TDO) is a graphical user interface (GUI) operating in the Windows 95/98/NT environment that allows the student to obtain solutions to optimization problems of the form (2.1). It was developed using Visual C++. The main window of TDO is shown in Figure 3. This Main window is used to control the whole optimization process, which includes the specification of the objective function (analytical or approximated), the specification of the design variable names, initial values and/or bounds, the specification of the constraints, and the optimization algorithm settings. After each item has been set or selected, control returns to this main window, from where the solution of the optimization problem is launched. This central control location allows the user to easily compare different algorithms, and to determine the influence of different settings, e.g., bounds and move limits. The current version of TDO is limited to five design variables, and the specification of three equality and three inequality analytical constraint functions. The approximation of the objective function and/or one constraint function is allowed for.

4.2 Specification of Analytical Functions TDO allows the user to specify analytical functions in terms of design variables. Several built-in analytical functions are provided. These are mainly selected through the specification of the coefficients of polynomials and reciprocal terms. The following general analytical objective function is included in the current version of TDO:

$$
\begin{aligned}
f(\mathbf{x}) = {} & a_{00} + a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + a_{41}x_4 + a_{51}x_5 \\
& + a_{12}x_1^2 + a_{22}x_2^2 + a_{32}x_3^2 + a_{42}x_4^2 + a_{52}x_5^2 \\
& + a_{13}x_1^3 + a_{23}x_2^3 + a_{33}x_3^3 + a_{43}x_4^3 + a_{53}x_5^3 \\
& + a_{14}x_1^4 + a_{24}x_2^4 + a_{34}x_3^4 + a_{44}x_4^4 + a_{54}x_5^4 \\
& + ac_{12}x_1x_2 + ac_{13}x_1x_3 + ac_{14}x_1x_4 + ac_{15}x_1x_5 + ac_{23}x_2x_3 + ac_{24}x_2x_4 + ac_{25}x_2x_5 \\
& + ac_{34}x_3x_4 + ac_{35}x_3x_5 + ac_{45}x_4x_5 \\
& + \frac{ar_1}{x_1} + \frac{ar_2}{x_2} + \frac{ar_3}{x_3} + \frac{ar_4}{x_4} + \frac{ar_5}{x_5} \\
& + as_1x_1^2x_2 + as_2x_1x_2x_3
\end{aligned} \tag{4.1}
$$

By specifying the $a_{i,j}$, $ac_{i,j}$, ar_i and as_i coefficients, the user can select any specific function from the set defined by (4.1). Of interest to the student is that transcendental and hyperbolic functions can also be approximated by polynomial functions, and can thus also be specified using (4.1). Refer to Figure 4 for the objective function dialog. The ‘Approximated’ setting is discussed in Section 4.4.

The settings of the coefficients of the analytical objective function, as selected by the ‘Set Coefficients’ button, are defined in the dialog contained in Figure 5. Note that the settings can be reset when different problems are run in succession. For constrained optimization problems, selecting ‘Constraints’ in the main window, allows for the specification of inequality constraint functions $g_j(\mathbf{x})$, and the equality constraint functions $h_j(\mathbf{x})$ in an identical manner to that shown for $f(\mathbf{x})$ in (4.1) and Figure 5.

4.3. Specification of Design Variables and Optimization Settings Returning to the main window, the selection of the ‘Design Variables’ setting calls the Design Variable window which enables the user to name variables and to select their initial values as shown in Figure 6. Any variable name can be given in the appropriate edit boxes. Constraints in the form of bounds can be placed on the variables through the selection of the appropriate check boxes and the specification of the relevant minimum and maximum values.

Selecting ‘Algorithm Settings’ in the main window enables the selection of a suitable optimization algorithm, ETOP, SQSD or LFOP for unconstrained problems, LFOPC or ETOPC for constrained problems and DYNAMIC-Q if approximations are to be employed. The window that allows for the appropriate selections is depicted in Figure 7. For each of these methods, the user can specify the convergence parameters (Design Variable Tolerance and Objective Function Gradient Norm) as well as a step size limit linked to the dimension of the design variable vector and range of the design variables. Default values of the algorithm control parameters are displayed in the dialog. These values are used if not modified. The maximum number of iterations and the print frequency of the results can also be adjusted in this dialog.

4.4. Optimization using approximations TDO uses successive spherical quadratic approximations (Snyman et al. 1994 and Snyamn and Hay 2000b) of the objective and one constraint function for cases where these functions, evaluated externally to TDO, are expensive to evaluate. The construction of the approximations at a local design point $\mathbf{x}^{(k)}$ requires the function value and its gradient at this current design point. The gradient of the function is obtained by first-order forward finite differences. If the ‘Approximated’ setting is selected TDO requires that the user enter the value of the relevant function at $\mathbf{x}^{(k)}$ as well as each of the values at the respective perturbation points $(\mathbf{x}^{(k)} + \mathbf{D}\mathbf{x}_i)$, i=1,n. Refer to Figure 8 for the dialog for the Approximated

Subproblem Setup. In the case of the first iteration, the checkbox for the first iteration is checked and the initial curvature can be specified (positive for a convex and negative for a concave approximation). A default value of 0.0 is used for the curvature (i.e. a linear initial approximation) if not changed. The solution of successive sub-problems is controlled from this dialog. For the first iteration, the initial design (starting values of the design variables) as well as the perturbations on the design variables, and move limits on design variable modification, are first specified. The objective and/or constraint function values, as obtained from an external numerical (or experimental) simulation, are entered in the Current Iteration fields. After a sub problem is solved, the previous objective function and design variables' values are automatically written in the Previous Iteration fields, and the design suggested by the optimizer automatically becomes the new Starting values of the design variables for the 'new' current iteration. The user then reruns the external simulation with the new design and the cycle is repeated.

4.5. Results of optimization problem The summary results of the direct solution of an analytical optimization problem, or of each sub problem using approximations, are given in the Results dialog shown in Figure 9. This window gives the results for the cable configuration problem with m_1=m_2 =1kg and $\ell_1 = \ell_2 = 1$m. (As a matter of interest the computed solution for the beer can problem with V_{spec}=400 cm^3 is x_1 =7.97885 cm and x_2= 8.00000 cm). The detailed results are written to a file that can be imported into a spreadsheet program (Microsoft Excel) for graphical output by clicking on the 'View history in Microsoft Excel' button. A macro in Excel reads the data and plots the history of the objective function, design variables and constraints. The numerical data values are also given in spreadsheet format for further processing. Alternatively, the user can click on the 'Graphical Display' button to view the results in a plot inside TDO. This view is shown in Figure 10. The objective function, constraints and design variables are shown on the same axis, and are normalized. The normalization factors are given in the dialog for all the functions and variables.

5. Implementation of TDO in a Design Course

Over the past few years TDO has successfully been used in the teaching of optimization techniques to relatively large groups and to individual students. In particular, it was recently employed in a senior design course for engineering students where the following assignment was set:

"Assignment: Introductory mathematical modeling and optimization exercise using TDO.

*This assignment represents a challenge to your creativity. Construct an original model of a real-world problem situation, simple enough (with respect to the forms of the objective and constraint functions and the number of variables) to be solved by TDO. In the formulation identify the parameters **p** of the model and the design variables **x**. Obtain a realistic solution to the problem by executing the modeling-optimization loop* (Figure 1) *as many times as necessary."*

With very little formal knowledge of mathematical optimization algorithms, the students were capable of solving a wide range of realistic modeling problems. The problems ranged from the optimal design of amplifiers, filters and antennas of importance to electrical engineers, to design problems relating to combustion chambers, centrifuges, cycle chains and formula 1 GP racers of specific interest to mechanical engineers. Many other problems were also successfully solved. Some of those worthy of further mentioning include the design of a solid rocket fuel projectile, optimizing the flow in a continuous casting process, the shape optimization of a soap bar for longer life and the design of a feeding trough for animals. Most of the problems tackled involved three to five variables, with many side constraints and relatively complicated inequality and equality constraints. In solving these non-trivial design problems, the students had to consider many different possible models (by varying, for example, the set of parameters **p**), and solving for each model the associated optimization problem. These tasks the students accomplished with remarkable ease, mainly due to the availability of TDO's user-friendly GUI, through which the models could easily be modified and optimized.

The above teaching experience has shown that, by giving the student assistance in the detailed, laborious and repetitive optimization task, enables him or her not only to solve non-trivial design problems, but also to have fun in doing so. As summarized by one of the students: *"The TDO program is a fun and useful tool in learning design optimization practice."*

Future possible improvements in TDO include the automatic linking of TDO to other simulation software to evaluate the objective and constraint functions. Other considerations are the extension to a considerably larger number of design variables; the approximation of multiple inequality and equality constraints; and the availability of a wider class of built-in analytical objective and constraint functions.

REFERENCES

-Arora, J.S., 1989, *Introduction to optimum design*, McGraw-Hill, New York.

-Snyman, J.A., 1982, "A new and dynamic method for unconstrained minimization", *Appl. Math. Modelling*, **6**, 449-462.

-Snyman, J.A., 1983, "An improved version of the original leap-frog dynamic method for unconstrained minimization LFOP1(b)", *Appl. Math. Modelling*, **7**, 216-218.

-Snyman, J.A., 1985 "Unconstrained minimization by combining the dynamic and conjugate gradient methods", *Quaestiones Mathematicae*, **8**, 33-42.

-Snyman, J.A., 1998, "ETOPC: A FORTRAN program for solving general constrained minimization problems by the conjugate gradient method without explicit line searches", *Research Report*, Department of Mechanical Engineering, University of Pretoria.

-Snyman, J.A., 2000, "The LFOPC leap-frog method for constrained optimization, *Computers Math. Applic*., **40**, 1085-1096.

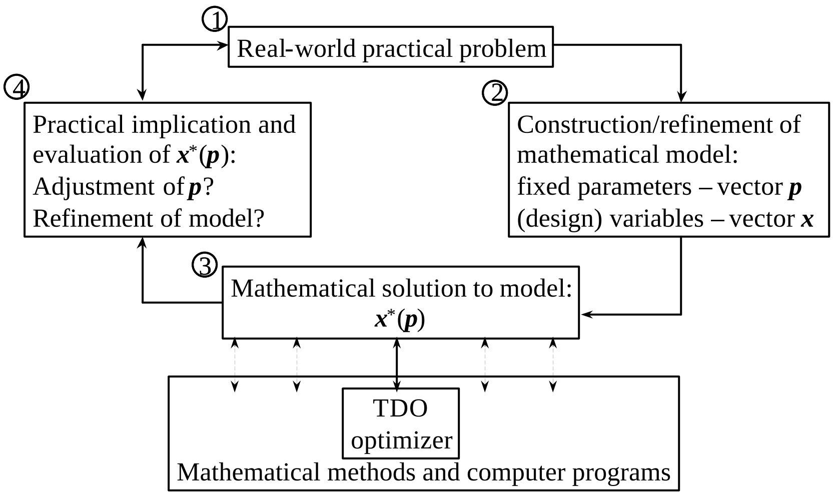
-Snyman, J. A., De Kock, D. J., Craig, K. J., Venter, P. J., 2001, "Toolkit for Design Optimization (TDO): An educational aid to mathematical modeling and optimization", *Quaestiones Mathematicae*, Suppl.**1**, 227-236.

-Snyman, J.A, Hay, A.M., 2000a, "The spherical quadratic steepest descent method for unconstrained minimization with no explicit line searches", *Computers Math. Applic*., **42**, 169-178.

-Snyman, J.A., Hay, A.M., 2000b, "The Dynamic-Q Optimization method: an alternative to SQP?, Proceedings of the International Workshop on Multidisciplinary Design Optimization, University of Pretoria, Pretoria, South Africa, 163-172.

-Snyman, J.A., Stander N., Roux, W.J., 1994, "A dynamic penalty function method for the solution of structural optimization problems", *Appl. Math. Modelling*, **18**, 453-460.

Figure 1. The mathematical modeling process.



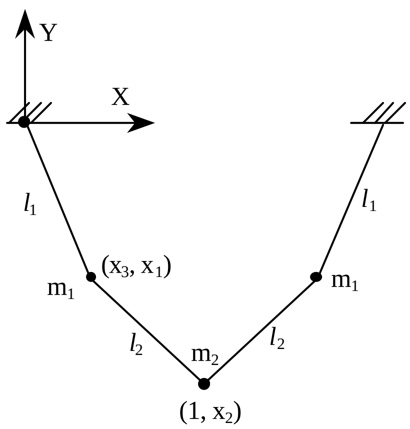


Figure 2. Cable configuration problem.

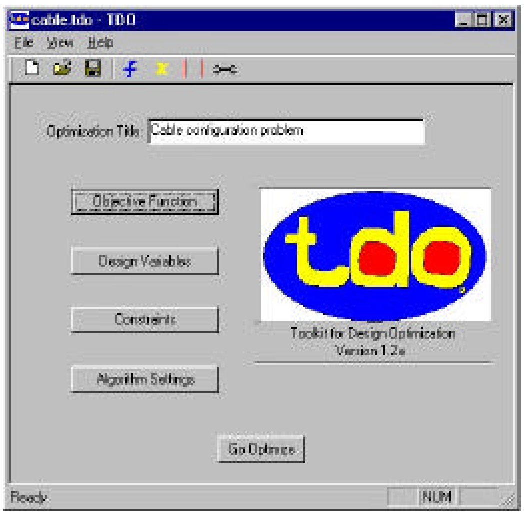


Figure 3. Main window of TDO.

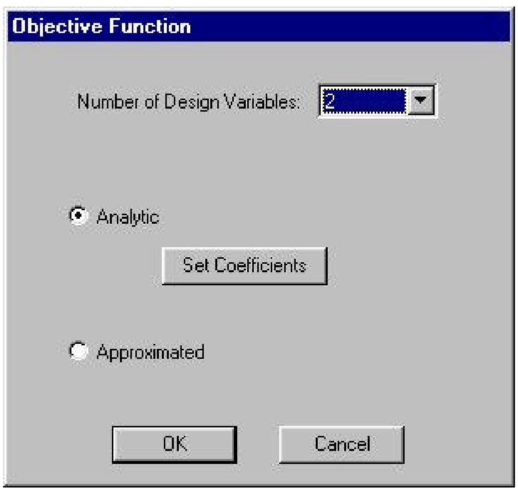


Figure 4. Objective Function dialog for beer can problem.

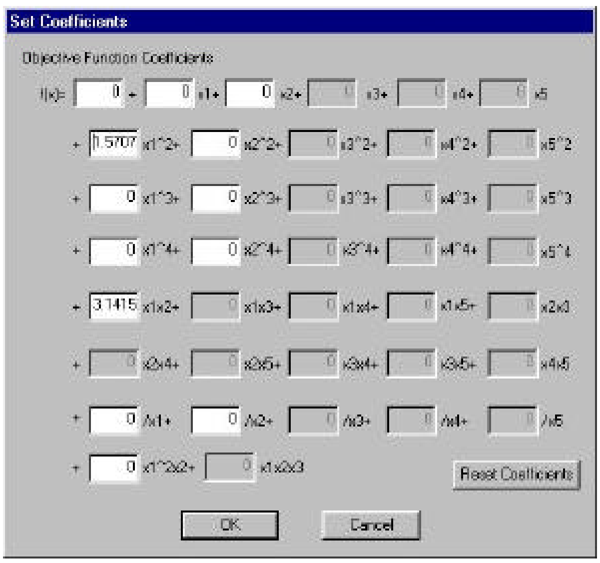


Figure 5. Analytic Objective Function Coefficient

settings dialog for beer can problem.

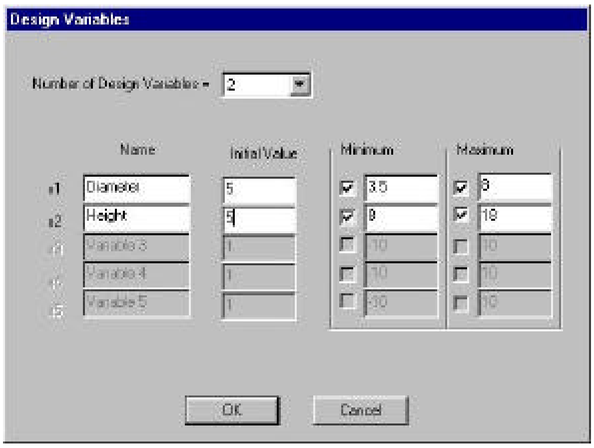


Figure 6. Design Variables dialog for beer can problem.

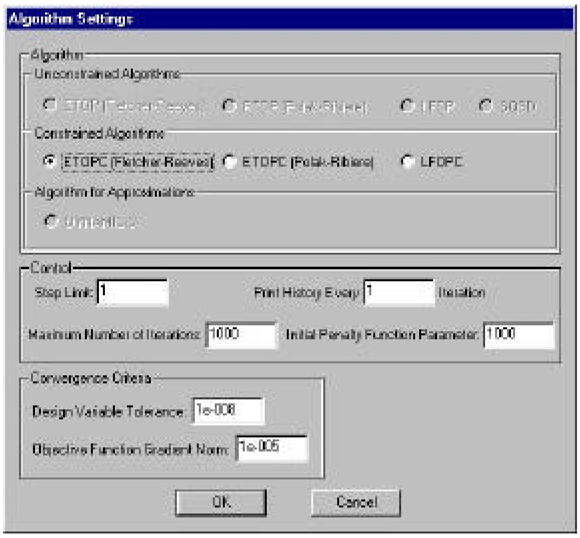


Figure 7. Optimization settings dialog for beer can problem.

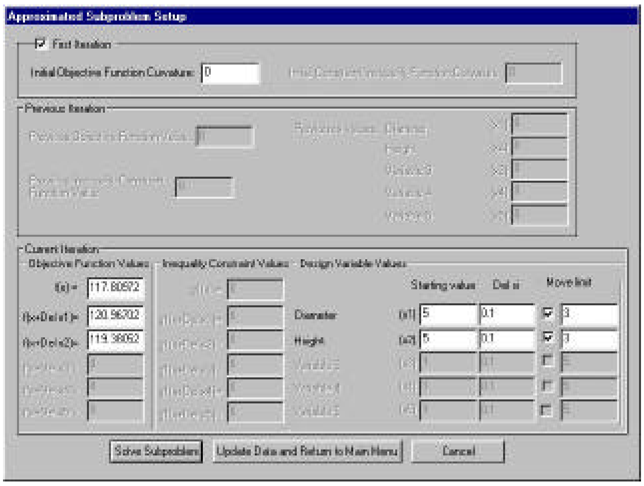


Figure 8. Approximations dialog.

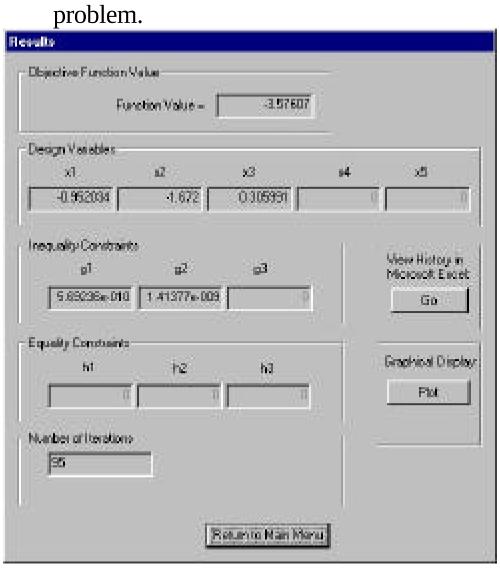


Figure 9. Results dialog for cable configuration problem.

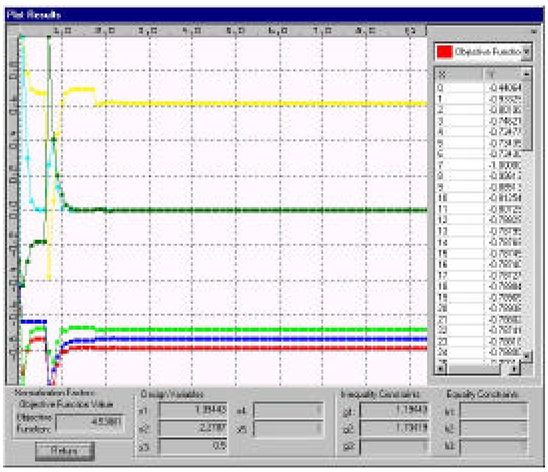


Figure 10. Graphical convergence histories of the values of the objective and constraint functions, and of the values of the design variables.